

В. Г. Скобелев

**Локальные
Алгоритмы
на
Графах**

ИПММ НАНУ

НАЦИОНАЛЬНАЯ АКАДЕМИЯ НАУК УКРАИНЫ
ИНСТИТУТ ПРИКЛАДНОЙ МАТЕМАТИКИ И МЕХАНИКИ

В.Г. Скобелев

**ЛОКАЛЬНЫЕ АЛГОРИТМЫ
НА ГРАФАХ**

Донецк

2003

УДК 519.15+681.14

Под редакцией

Д.т.н., заведующего отделом теории управляющих систем Института прикладной математики и механики Национальной Академии Наук Украины *В.Н. Ткаченко*

Рецензенты:

Профессор, д.т.н., декан факультета компьютерных наук и информационных технологий Саратовского государственного университета им. Н.Г. Чернышевского *Д.В. Сперанский*

Академик РАЕН, профессор, д.т.н., проректор по информационным технологиям и открытому образованию Саратовского государственного социально-экономического университета *А.А. Сытник*

Member of the Estonian Academy of Sciences, Professor, Dr. Sc., Professor of Tallinn Technical University *Raimund Ubar*

Локальные алгоритмы на графах.

В.Г. Скобелев. ИПММ НАН Украины, Донецк, 2003. – 217с.

ISBN 966-02-2987-9

Монография посвящена исследованию разрешимости задач теории графов в классах алгоритмов с линейной емкостью сложностью и алгоритмов с линейной сложностью рабочей памяти. Для различных представлений графов исследуется сложность операций над графами, а также сложность решения задач построения всех основных типов путей, циклов и остовных деревьев. Показано, что ряд модельных задач дискретной математики и ее приложений (идентификация состояний конечного автомата, построение супервизора для автоматной модели системы дискретных событий, построение выигрышной стратегии в игре 2-х лиц на графе) не разрешим в рассматриваемых классах алгоритмов.

Для специалистов в областях дискретной математики, теории графов и computer science, студентов и аспирантов, специализирующихся в этих областях, а также для специалистов, чьи исследования связаны с разработкой и анализом алгоритмов как в теоретическом, так и в прикладном аспектах.

Утверждено к печати Ученым советом Института прикладной математики и механики НАН Украины

Локальні алгоритми на графах.

В.Г. Скобелев. ИПММ НАН України, Донецьк, 2003. – 217с. (Російською мовою).

ISBN 966-02-2987-9

Монографія присвячена дослідженню можливості вирішення задач теорії графів у класах алгоритмів, що мають лінійну ємність, або лінійну ємність робочої пам'яті. Для різних представлень графів досліджена складність операцій над графами, та складність вирішення задач побудови усіх базових типів шляхів, циклів, та остовних дерев. Встановлено, що є деякі модельні задачі дискретної математики, та її застосувань (ідентифікація станів фінітних автоматів, розробка супервизора для системи дискретних подій, що представлена у вигляді автоматної моделі, розробка вигранної стратегії для гри двох гравців на графі), що не можуть бути вирішені за допомогою запропонованих алгоритмів.

Для спеціалістів у галузях дискретної математики, теорії графів, та computer science, студентів та аспірантів, що спеціалізуються у цих галузях, а також для спеціалістів, що займаються розробкою та аналізом алгоритмів, як з позиції теорії, так і з позиції їх застосувань.

ISBN 966-02-2987-9

© В.Г. Скобелев

ПРЕДИСЛОВИЕ

Предмет исследования монографии – анализ задач теории графов с позиции их разрешимости в классах простейших алгоритмов, а именно: алгоритмов с линейной емкостной сложностью, а также алгоритмов с линейной сложностью рабочей памяти.

Актуальность выбранной тематики обусловлена следующими обстоятельствами. Граф является одним из основных объектов, используемых в качестве математической модели для широкого класса прикладных задач. По этой причине исследование сложности решения задач теории графов имеет большое значение, как для теории графов, так и при разработке и анализе алгоритмов. Однако в теории графов преобладают дескриптивный и метрический подходы, а алгоритмическому подходу уделяется недостаточное внимание. Аналогичным образом, в теории алгоритмов преобладает исследование временной сложности, а многие вопросы, связанные с анализом емкостной сложности остаются в тени.

Монография состоит из четырех разделов.

Раздел 1 представляет собой развернутое введение. В нем изложен необходимый математический аппарат. Проведен ретроспективный анализ предмета исследования, сформулированы цели и показана их актуальность.

В разделе 2 исследуется сложность выполнения операций и проверки отношений (типа изоморфное включение) над обычными и направленными графами. Проведен сравнительный анализ этой сложности для представлений (и обычных, и направленных) графов, как списками, так и матрицами смежности (как вершин, так и ребер (для графов) и дуг (для направленных графов)). Получены оценки ряда свойств почти всех графов, а также ряд оценок, связанных со «сложностью в среднем».

В разделе 3 систематически исследуется разрешимость задач построения основных типов путей, циклов и остовных деревьев в классе предложенных алгоритмов. Получен ряд нижних оценок функций Шеннона, характеризующих количества этих объектов в графах, имеющих заданный порядок и размер. Достаточное внимание уделено исследованию (основанного на аксиоматике PERT и имеющего многочисленные применения при решении прикладных задач) процесса *pebbling* – специального обхода ациклического направленного орграфа с использованием «камешков» для покрытия посещаемых вершин.

В разделе 4 показана неразрешимость в классе предложенных алгоритмов ряда модельных задач (идентификация внутренних состояний конечного автомата, построение супервизора для системы дискретных событий автоматного типа, построение выигрышной стратегии для ряда игр двух лиц на графах).

Монография написана в замкнутой форме, т.е. определяются все понятия, кроме общепринятых.

Полученные результаты – оригинальные результаты автора – старшего научного сотрудника отдела теории управляющих систем ИПММ НАНУ и отражают тематику исследований отдела. Часть результатов, представленных в пп. 3.5, 4.3 и 4.4, получена автором в рамках исследований по теме «Логический подход к управлению динамическими системами», Рег. № 01024004565.

Монография предназначена для специалистов в областях дискретной математики и computer science, студентов и аспирантов, специализирующихся в этих областях, а также специалистов, занимающихся разработкой и анализом алгоритмов, как в теоретическом, так и прикладном аспекте. Она также может быть использована преподавателями при разработке спецкурсов для специальностей прикладная математика и computer science.

Благодарности. Непосредственным стимулом к написанию данной монографии послужил ряд лекций по теории графов, которые профессор *Зыков А.А.* прочитал весной 2002 года в ИПММ НАНУ, за что автор считает своим долгом выразить ему искреннюю благодарность. За возможность обсуждать задачи в процессе их постановки и решения автор благодарит своих коллег по отделу и институту.

Автор выражает искреннюю благодарность администрации ИПММ НАНУ – директору, академику НАН Украины, профессору *И.В. Скрыпнику* и зам. директора, члену-корреспонденту НАН Украины, профессору *А.М. Ковалеву* за обеспечение идеальных условий для работы над монографией.

Написание любой книги требует больших затрат сил и времени. Автор выражает глубокую и искреннюю благодарность своим близким – жене *Галине* и сыну *Владимиру* за их терпение и постоянную поддержку в процессе работы над монографией.

Безусловно, что за все недостатки ответственность несет только автор.

Автор будет благодарен за любые конструктивные замечания, касающиеся содержания книги.

АВТОР

Декабрь, 2001, Донецк

СОДЕРЖАНИЕ

СПИСОК ОБОЗНАЧЕНИЙ	7
1. Модели и методы	11
1.1. <i>Машины Тьюринга</i>	11
1.2. <i>Графы</i>	21
1.3. <i>Автоматы в лабиринтах</i>	35
1.4. <i>Основные понятия и определения</i>	45
1.5. <i>Выводы</i>	79
2. Анализ представлений графов	81
2.1. <i>Предварительные замечания</i>	82
2.2. <i>Представления графов списками смежности</i>	84
2.3. <i>Представление направленных графов списками смежности</i>	109
1.4. <i>Выводы</i>	125
3. Классы $Lcl(\mathbf{R})$ и $WORK - Lcl(\mathbf{R})$	127
3.1. <i>Предварительные замечания</i>	127
3.2. <i>Задачи построения путей в графе</i>	130
3.3. <i>Задачи построения циклов в графе</i>	147
3.4. <i>Построение остовных деревьев</i>	159
3.5. <i>Rebbling</i>	167
3.6. <i>Выводы</i>	180

4. За пределами класса $WORK - Lcl(\mathbf{R})$	181
1.1. <i>Предварительные замечания</i>	181
1.2. <i>Задачи идентификации состояний автоматов</i>	183
1.3. <i>Системы дискретных событий</i>	193
1.4. <i>Игры на графах</i>	205
1.5. <i>Выводы</i>	213
Заключение	215
СПИСОК ЛИТЕРАТУРЫ	217

СПИСОК ОБОЗНАЧЕНИЙ

Логические:

$A \Rightarrow B$ – если A , то B ;

$A \Leftrightarrow B$ – A тогда и только тогда, когда B ;

\bar{a} ($\neg a$) – отрицание a , т.е. $\bar{a} = 1$ тогда и только тогда, когда $a = 0$;

$a \wedge b$ ($a \& b$) – конъюнкция a и b , т.е. $a \wedge b = 1$ тогда и только тогда, когда $a = b = 1$;

$a \vee b$ – дизъюнкция a и b , т.е. $a \vee b = 0$ тогда и только тогда, когда $a = b = 0$;

$(\forall x)P(x)$ – для всех x утверждение $P(x)$ – истинное;

$(\exists x)P(x)$ – существует такое x , что утверждение $P(x)$ – истинное.

Теоретико-множественные:

\emptyset – пустое множество, т.е. множество, не содержащее ни одного элемента;

$x \in X$ – x является элементом множества X ;

$x \notin X$ – x не является элементом множества X ;

$X = Y$ – множества X и Y равны, т.е. состоят из одних и тех же элементов;

$X \subseteq Y$ – множество X является подмножеством множества Y , т.е. каждый элемент множества X является элементом множества Y ;

$X \not\subseteq Y$ – множество X не является подмножеством множества Y ;

$X \subset Y$ – $X \subseteq Y$ и существует такой элемент $x \in X$, что $x \notin Y$;

$X \not\subset Y$ – утверждение $X \subset Y$ – ложное;

$\{x | P(x)\}$ – множество всех таких x , что утверждение $P(x)$ – истинное;

$\mathbf{B}(X)$ – булеан множества X , т.е. множество всех подмножеств множества X ;

$|X|$ – мощность множества X ;

\aleph_0 – мощность счетного множества;

$X \cap Y$ – пересечение множеств X и Y , т.е. $x \in X \cap Y$ тогда и только тогда, когда $x \in X$ и $x \in Y$;

$X \cup Y$ – объединение множеств X и Y , т.е. $x \in X \cup Y$ тогда и только тогда, когда $x \in X$ или $x \in Y$;

$X \setminus Y$ – разность множеств X и Y , т.е. $x \in X \setminus Y$ тогда и только тогда, когда $x \in X$ и $x \notin Y$;

X / ε – фактор-множество множества X по отношению эквивалентности ε ;

$X \times Y$ – декартово произведение множеств X и Y , т.е. $(x, y) \in X \times Y$ тогда и только тогда, когда $x \in X$ и $y \in Y$;

\mathbf{N} – множество всех натуральных чисел;

\mathbf{Z} – множество всех целых чисел;

\mathbf{Z}_+ – множество всех неотрицательных целых чисел;

\mathbf{Z}_k ($k \in \mathbf{N}$) – множество, состоящее из чисел $0, 1, \dots, k-1$;

\mathbf{R} – множество всех действительных чисел;

\mathbf{R}_+ – множество всех неотрицательных действительных чисел;

\mathbf{X}^n ($n \in \mathbf{N}$) – n -я декартова степень линейного пространства \mathbf{X} , т.е. множество всех таких векторов (x_1, \dots, x_n) , что $x_1, \dots, x_n \in \mathbf{X}$.

Функциональные:

$f: X \rightarrow Y$ – (возможно частичное) отображение множества X в множество Y ;

$Dom f$ – область определения отображения f ;

$Val f$ – множество значений отображения f ;

$\ker f$ – ядерная эквивалентность отображения f .

Асимптотические:

$f(n) = o(g(n))$ ($n \rightarrow \infty$) – если $n \rightarrow \infty$, то функция $f(n)$ представляет собой “о” малое от функции $g(n)$, т.е. $\lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} = 0$;

$f(n) = O(g(n))$ ($n \rightarrow \infty$) – если $n \rightarrow \infty$, то функция $f(n)$ представляет собой “о” большое от функции $g(n)$, т.е. существует такое значение $n_0 \in \mathbf{N}$ и такая константа $c > 0$, что $f(n) \leq c g(n)$ для всех $n > n_0$ ($n \in \mathbf{N}$);

$f(n) = \Omega(g(n))$ ($n \rightarrow \infty$) – если $n \rightarrow \infty$, то функция $f(n)$ представляет собой “омега” большое от функции $g(n)$, т.е. существует такое значение $n_0 \in \mathbf{N}$ и такая константа $c > 0$, что $f(n) \geq c \cdot g(n)$ для всех $n > n_0$ ($n \in \mathbf{N}$);

$f(n) = \Theta(g(n))$ ($n \rightarrow \infty$) – если $n \rightarrow \infty$, то функция $f(n)$ представляет собой “тэта” большое от функции $g(n)$, т.е. существует такое значение $n_0 \in \mathbf{N}$ и такие положительные константы c_1 и c_2 ($c_1 > c_2$), что $c_1 \cdot g(n) \geq f(n) \geq c_2 \cdot g(n)$ для всех $n > n_0$ ($n \in \mathbf{N}$);

$f(n) \sim g(n)$ ($n \rightarrow \infty$) – если $n \rightarrow \infty$, то функции $f(n)$ и $g(n)$ асимптотически эквивалентны, т.е. $\lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} = 1$.

Лингвистические:

Λ – пустое слово, т.е. не содержащее ни одного символа;

X^n ($n \in \mathbf{N}$) – множество всех слов длины n в алфавите X ;

X^+ – множество всех непустых слов в алфавите X ;

X^* – множество всех, включая и пустое, слов в алфавите X .

Матричные:

\mathbf{I}_n – единичная матрица n -го порядка;

$\mathbf{A} + \mathbf{B}$ – сумма матриц \mathbf{A} и \mathbf{B} ;

$\mathbf{A} \cdot \mathbf{B}$ – произведение матриц \mathbf{A} и \mathbf{B} ;

\mathbf{A}^{-1} – матрица, обратная матрице \mathbf{A} ;

\mathbf{A}^T – матрица, транспонированная к матрице \mathbf{A} ;

$|\mathbf{A}|$ – определитель матрицы \mathbf{A} ;

$\overline{\mathbf{A}}$ – поэлементное дополнение матрицы \mathbf{A} ;

$\mathbf{A} \wedge \mathbf{B}$ – поэлементная конъюнкция булевых матриц \mathbf{A} и \mathbf{B} ;

$\mathbf{A} \vee \mathbf{B}$ – поэлементная дизъюнкция булевых матриц \mathbf{A} и \mathbf{B} ;

$\mathbf{A} \oplus \mathbf{B}$ – поэлементная сумма по модулю 2 булевых матриц \mathbf{A} и \mathbf{B} ;

$\mathbf{A} \otimes \mathbf{B}$ – произведение булевых матриц \mathbf{A} и \mathbf{B} ;

\mathbf{A}^l ($l \in \mathbf{Z}_+$) – l -я булева степень матрицы \mathbf{A} , т.е. $\mathbf{A}^0 = \mathbf{I}$, $\mathbf{A}^1 = \mathbf{A}$ и $\mathbf{A}^{\otimes l} = \underbrace{\mathbf{A} \otimes \dots \otimes \mathbf{A}}_{l \text{ раз}}$ для всех $l \geq 2$ ($l \in \mathbf{N}$).

Стохастические:

$\mathbf{M}\xi$ – математическое ожидание дискретной случайной величины ξ , т.е. если x_1, \dots, x_n значения дискретной случайной величины ξ , принимаемые, соответственно, с вероятностями p_1, \dots, p_n , то $\mathbf{M}\xi = \sum_{i=1}^n x_i \cdot p_i$;

$\mathbf{D}\xi$ – дисперсия дискретной случайной величины ξ , т.е. $\mathbf{D}\xi = \mathbf{M}\xi^2 - (\mathbf{M}\xi)^2$.

1. МОДЕЛИ И МЕТОДЫ.

Значительное число существенно отличающихся друг от друга объектов обладают естественной *сетевой структурой*. Поэтому язык теории графов имеет исключительные возможности, а его наглядность, делает доступными формулировки достаточно сложных задач и весьма тонкие методы их решения. Как следствие, *построение и анализ алгоритмов на графах* составляет основу исследования широкого класса задач дискретной математики и ее многочисленных приложений (таких, как анализ и проектирование систем, генетика, кристаллография, лингвистика, социальные науки, химия, экономика, электротехника и т.д.). Значение этих алгоритмов существенно возросло с тех пор, как *компьютерное моделирование* стало нормой (а, часто, и единственным методом) решения сложных фундаментальных и прикладных задач. Анализ сложности алгоритмов на графах (если он вообще производится) часто сводится к асимптотической оценке *временной сложности* в терминах числа вершин (иногда и/или ребер графа). Безусловно, такой подход дает возможность охарактеризовать исследуемый алгоритм на графе в терминах классической теории алгоритмов. При этом часто не принимается во внимание тот фактор, что граф может быть *представлен* различными способами, причем сложность различных представлений – различная. Более того, для различных представлений выделяются различные классы *элементарных операций* на графе (сложность которых, в свою очередь, зависит от выбранного представления). Поэтому естественно зафиксировать одно из наиболее распространенных компьютерно-ориентированных представлений графа и классифицировать алгоритмы на графах в терминах *емкостной сложности* именно этого представления. Ясно, что такая классификация алгоритмов не совпадает с общепринятой классификацией. Однако она дает возможность естественно выделить *локальные алгоритмы на графах*, т.е. такие, которые могут быть реализованы в пределах памяти, объем которой асимптотически равен емкостной сложности выбранного представления графа. Именно эти алгоритмы и являются предметом исследования настоящей монографии.

В настоящем разделе изложен необходимый математический аппарат. П.1.1. посвящен машине Тьюринга – наиболее распространенной формализации интуитивного понятия *алгоритм*. Эта модель дает возможность в процессе анализа сложности алгоритмов легко переходить от равномерного веса к логарифмическому весу. П.1.2. содержит основную терминологию теории графов. В п.1.3 представлены основные результаты, связанные с исследованием поведения автоматов в лабиринтах – раздела дискретной математики, устанавливающего тесную связь между теорией алгоритмов и теорией графов. В п.1.4. введены необходимые понятия и определения и очерчен предмет исследования. П.1.5 содержит краткие выводы.

1.1. Машины Тьюринга.

Машина Тьюринга (МТ) – это формальная модель интуитивного понятия алгоритм. Содержательно, (детерминированная) МТ состоит из *лент, головок*, каждая из которых связана с единственной лентой и *устройств* с конечным числом состояний, *управляющих* головками. Каждая головка управляется единственным управляющим устройством, а каждая лента – это *целочисленная решетка* типа \mathbf{Z}^n или \mathbf{Z}_+^n ($n \in \mathbf{N}$), элементы которой называются *клетками*. Известны различные варианты МТ, отличающиеся друг от друга по типу и числу лент, числу головок и числу

управляющих устройств. Однако, во всех случаях МТ – это упорядоченная семерка объектов

$$M = (Q, A, I, \delta, \Lambda, q_{in}, q_{fin}),$$

где Q – конечное множество *состояний* управляющих устройств, A – конечный *внешний алфавит*, т.е. множество символов, которые могут быть записаны в клетках ленты, I ($\emptyset \neq I \subseteq A$) – *входной алфавит*, Λ ($\Lambda \in A \setminus I$) – *пустой символ*, $q_{in} \in Q$ и $q_{fin} \in Q$ – соответственно, *начальное* и *финальное состояние* управляющего устройства, а δ – (возможно, частичная) *функция переходов*.

Функционирование МТ осуществляется в дискретные моменты времени, т.е. в моменты $t \in \mathbf{N}$, в соответствии с функцией переходов δ . В начальный момент времени все клетки лент, за исключением быть может их конечного числа, содержат пустой символ Λ , каждая головка *обозревает* некоторую фиксированную клетку на соответствующей ленте, а каждое управляющее устройство находится в начальном состоянии q_{in} . *Такт* работы МТ состоит в следующем. Действия каждой головки определяются состоянием $q_1 \in Q$ управляющего этой головкой устройства и записанным в обозреваемой ею клетке символом $a_1 \in A \cup \{\Lambda\}$ (т.е. парой (q_1, a_1)) и осуществляются (в соответствии с функцией переходов δ) следующим образом. Содержимое обозреваемой клетки заменяется символом $a_2 \in A \cup \{\Lambda\}$, а головка либо сдвигается в соседнюю клетку, т.е. имеющую с обозреваемой клеткой *общую границу*, либо остается в той же клетке. Управляющее головкой устройство в течение этого же такта переходит в состояние $q_2 \in Q$. Допускается, что $a_2 = a_1$ и/или $q_2 = q_1$. Если хотя бы одно управляющее устройство переходит в финальное состояние q_{fin} , то все дальнейшие действия прекращаются, и происходит *остановка* МТ.

Конфигурация МТ в момент времени t ($t \in \mathbf{N}$) – это отображение K_t , указывающее для каждой клетки каждой ленты записанный в ней символ и, если клетка обозревается головкой, состояние, в котором находится соответствующее управляющее устройство. Таким образом, для каждой *начальной* конфигурации K_1 МТ M порождает конечный или бесконечный *процесс*

$$K : K_1, K_2, \dots$$

ее *переработки*. Выше было отмечено, что δ – возможно, частичная функция переходов. Если процесс K содержит конфигурацию, в которой ни одно управляющее устройство МТ M не достигло финального состояния q_f , а для некоторой пары (q_1, a_1) действия соответствующей головки

не определены, то без ограничения общности можно считать, что происходит *заикливание* МТ M , которое можно интерпретировать как *бесконечный процесс* переработки *неопределенной конфигурации* в *неопределенную конфигурацию*. Если \mathbf{K} – конечный процесс, то говорят, что МТ M *применима* к конфигурации K_1 , а если \mathbf{K} – бесконечный процесс, то говорят, что МТ M *не применима* к конфигурации K_1 .

Сложность переработки конфигураций МТ характеризуется различными *мерами сложности*, являющихся *функциями Шеннона*. Отметим, что *техника следов*, которую разработали Барздинь Я.М., Трахтенброт Б.А., Фрейвалд Р.М., Rabin М.О. и Hennie F.G. (см., напр. [11]), дает возможность установить верхние оценки той или иной меры сложности через оценки значений остальных мер сложности. Наиболее часто используются *временная* и *емкостная* меры сложности, определяемые следующим образом.

Рассмотрим процесс

$$\mathbf{K} : K = K_1, K_2, \dots$$

переработки МТ M конфигурации K . *Сигнализирующие времена* $t_M(K)$ и *емкости* $v_M(K)$ процесса \mathbf{K} определяются следующим образом. Если МТ M не применима к конфигурации K , то $t_M(K)$ и $v_M(K)$ не определены. Предположим теперь, что МТ M применима к конфигурации K . Тогда $t_M(K)$ – это число конфигураций в процессе \mathbf{K} . *Активной зоной конфигурации* K_i называется объединение минимальных связных частей всех лент МТ M , содержащее обозреваемые головками клетки и все клетки, содержащие символы, отличные от Λ . *Активная зона процесса* \mathbf{K} определяется как объединение минимальных частей всех лент МТ M , содержащее активные зоны всех конфигураций процесса \mathbf{K} . Сигнализирующая емкости $v_M(K)$ равна *объему активной зоны процесса* \mathbf{K} , который можно определить двумя способами: либо как число клеток активной зоны процесса \mathbf{K} , либо как максимум по всем лентам чисел клеток ленты, входящих в активную зону процесса \mathbf{K} .

С каждой начальной конфигурацией K МТ M можно сопоставить *размер* $size(K) \in \mathbf{N}$ определяемого ею *входа*. Временная и емкостная сложность переработки конфигураций МТ M определяются, соответственно, равенствами

$$t_M^{onp}(n) = \max_{size(K)=n} t_M(K),$$

$$v_M^{onp}(n) = \max_{size(K)=n} v_M(K) \quad (n \in \mathbf{N}).$$

Кроме функций $t_M(n)$ и $v_M(n)$, часто используются функции $t_M^*(n)$ и $v_M^*(n)$, определяемые равенствами

$$t_M^*(n) = \max_{1 \leq v \leq n}^{опр} t_M(v),$$

$$v_M^*(n) = \max_{1 \leq v \leq n}^{опр} v_M(v).$$

В процессе исследования сложности алгоритмов вместо функций $t_M(n)$ и $v_M(n)$ (соответственно, функций $t_M^*(n)$ и $v_M^*(n)$) часто используют их *асимптотические оценки* $T_M(n)$ и $V_M(n)$ (соответственно $T_M^*(n)$ и $V_M^*(n)$), т.е. функции вида

$$f(n) = O(g(n)) (n \rightarrow \infty),$$

что упрощает изложение и не нарушает общности рассуждений. Для асимптотической оценки сложности алгоритмов оба рассмотренных выше способа определения объема активной зоны процесса эквивалентны (т.к. число лент МТ M фиксировано).

Важное понятие, касающееся сложности алгоритмов, связано с мощностью внешнего алфавита A МТ. Если $|A \setminus \{\Lambda\}| = 2$ (в этом случае, как правило, полагают $A = \{0,1,\Lambda\}$), то говорят, что речь идет о *логарифмической весе*. Если же на мощность внешнего алфавита A не наложены никакие ограничения (за исключением того, что она конечна), то говорят, что речь идет о *равномерном весе*. Используя *кодирование* элементов множества $A \setminus \{\Lambda\}$ двоичными последовательностями (или, иными словами, *блоками*) длины

$$\tau = \lceil \log |A \setminus \{\Lambda\}| \rceil$$

(пустой символ Λ кодируется последовательностью $\underbrace{\Lambda \dots \Lambda}_{\tau \text{ раз}}$), *C. Shannon* в 1956 году доказал следующую теорему.

Теорема 1.1. МТ с внешним алфавитом A ($|A| \geq 3$) можно блочно моделировать МТ с внешним алфавитом $A_1 = \{0,1,\Lambda\}$ с логарифмическим замедлением (т.е. с растяжением во времени), пропорциональным величине $\lceil \log |A \setminus \{\Lambda\}| \rceil$.

Наиболее часто используются следующие две интерпретации процесса переработки конфигураций МТ.

Во-первых, это *вычисление значений (словарной) функции n ($n \in \mathbf{N}$) переменных*

$$y = f(x_1, \dots, x_n) \quad (x_1, \dots, x_n \in I^*).$$

Предполагается, что начальная конфигурация K содержит значения аргументов x_1, \dots, x_n , записанные в специально отведенных местах на лентах. Если $(x_1, \dots, x_n) \in \text{Dom } f$, то МТ применима к K , а конфигурация, полученная по завершению процесса переработки K , содержит записанное в специально отведенных клетках значение функции $f(x_1, \dots, x_n)$. Если же $(x_1, \dots, x_n) \notin \text{Dom } f$, то МТ не применима к конфигурации K . Используя методы композиций МТ, получившие впоследствии имя *Тьюрингово программирование* (см., напр. [11]), *A. Turing* в 1936 году доказал следующую теорему.

Теорема 1.2. Для каждой частично-рекурсивной функции $f(x_1, \dots, x_n)$ существует МТ, осуществляющая вычисление ее значений.

Во-вторых, это *распознавание языка L в алфавите I* . Предполагается, что начальная конфигурация K содержит слово $p \in I^*$, записанное в специально отведенных местах на лентах. Если $p \in L$, то МТ применима к K , а конфигурация, полученная по завершению процесса переработки K , содержит в специально отведенной клетке символ 1. Если же $p \notin L$, то либо МТ не применима к K , либо конфигурация, полученная по завершению процесса переработки K , содержит в специально отведенной клетке символ, не равный 1 (например, символ 0).

Если МТ M вычисляет такую функцию f , что $\text{Val } f = \{0,1\}$, то говорят, что M *вычисляет предикат f* . Таким образом, класс языков, распознаваемых МТ, определяется множеством тех (возможно, частичных) предикатов, которые могут быть вычислены МТ. Этот класс языков получил имя *рекурсивно-перечислимых*. Формально, множество слов $L \subseteq I^*$ – *рекурсивно-перечислимое*, если оно является множеством значений функции $f: \mathbf{N} \rightarrow I^*$, вычислимой некоторой МТ (здесь предполагается, что натуральные числа кодируются словами в алфавите I , так что f – словарная функция). Последнее означает, что существует МТ, которая, будучи запущенной на пустой ленте, генерирует бесконечную последовательность

$$1\# f(1)\#\#2\# f(2)\#\#3\# f(3)\#\#\dots\#\#n\# f(n)\dots\#\# \quad (\#\in E \setminus I),$$

т.е. перечисляет (возможно, с повторениями) элементы множества L .

Выше было отмечено, что существуют различные варианты МТ, отличающиеся друг от друга по типу и числу лент, числу головок и числу управляющих устройств. Рассмотрим кратко основные варианты МТ.

Обычная одноленточная МТ $M = (Q, A, I, \delta, \Lambda, q_{in}, q_{fin})$ с одной не ограниченной в обе стороны лентой изображена на рис. 1.1. Функция переходов такой МТ – это (возможно, частичное) отображение множества $Q \times A$

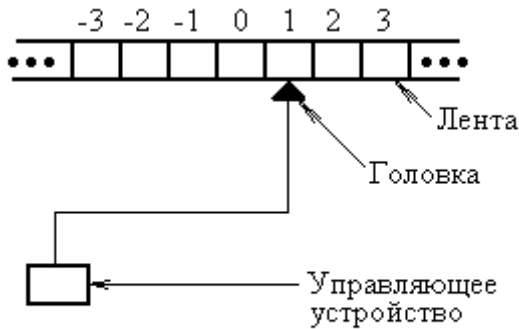


Рис. 1.1. Одноленточная МТ с одной не ограниченной в обе стороны лентой.

в множество $A \times Q \times \{L, P, H_m\}$ (L и P – сдвиг головки на одну клетку, соответственно, влево и вправо, а H_m – головка остается на месте). Отметим, что такую МТ всегда можно преобразовать в эквивалентную ей (т.е. вычисляющую ту же функцию или распознающую тот же язык) подвижную МТ, т.е. такую, что $Val \delta \subseteq A \times Q \times \{L, P\}$ (см., напр. [11]). Активная зона стандартной начальной конфигурации K расположена, начиная с 0-й клетки, вправо, а головка обозревает 0-ю клетку. Если МТ M применима к конфигурации K , то в стандартной заключительной конфигурации головка обозревает самую левую клетку, содержащую результат. Известно (см., напр. [11]), что если одноленточная МТ M с одной не ограниченной в обе стороны лентой применима к конфигурации K , то

$$t_M(K) \leq |A|^{v_M(K)} \cdot v_M(K) \cdot |Q|.$$

Обычная одноленточная МТ с одной односторонней лентой изображена на рис. 1.2. Единственное отличие этой модели от предыдущей состоит в том, что необходимо предотвратить схождение головки с ленты, когда она обозревает 0-ю клетку (т.е. запретить в этом случае сдвиг головки влево). Выход из этого положения состоит в том, что в 0-ю клетку записывается специальный маркер (например, #) – признак конца ленты. МТ с одной не ограниченной в обе стороны лентой можно преобразовать в эквивалентную МТ с одной односторонней лентой с не более чем константным растяжением сигнализирующей времени. Для этого достаточно использовать одностороннюю двухэтажную ленту (верхний этаж – для клеток, занумерованных неотрицательными числами, а нижний этаж – для клеток, занумерованных отрицательными числами (см., напр. [11])).

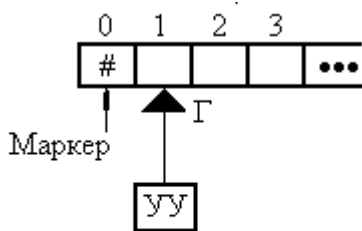


Рис. 1.2. Одноленточная МТ с одной односторонней лентой.

то

то

k -головочная одноленточная МТ $M = (Q, A, I, \delta, \Lambda, q_{in}, q_{fin})$ с одной не ограниченной в обе стороны лентой и h ($h \in \{1, \dots, k\}$) управляющими устройствами изображена на рис. 1.3. Функция переходов такой МТ – это (возможно, частичное) отображение множества $\times_{j=1}^h (Q \times A^{r_j})$ в множество $\times_{j=1}^h ((A \times \{L, P, NM\})^{r_j} \times Q)$.

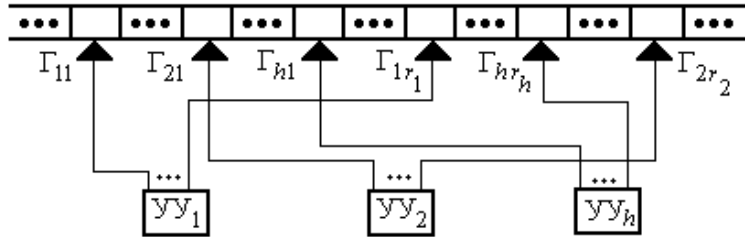


Рис. 1.3. k -головочная МТ с одной не ограниченной в обе стороны лентой и h управляющими устройствами ($r_1 + r_2 + \dots + r_h = k$).

Для такой МТ необходимо предотвратить *конфликтные ситуации*, возникающие, если несколько головок обозревают одну и ту же клетку, а функция переходов предписывает различным головкам осуществить в эту клетку различные записи. *Разрешение конфликта* состоит в том, что головки упорядочиваются по их приоритету, а запись в клетку осуществляет та из головок, обозревающих данную клетку, приоритет которой – наивысший. В 1956 году *Б.А. Трахтенброт* доказал следующую теорему.

Теорема 1.3. Для любой одноленточной многоголовочной МТ M существует такая эквивалентная ей обычная одноленточная МТ M_1 , что для любой конфигурации K , к которой применима МТ M , справедливо неравенство

$$t_{M_1}(K) \leq 3 \cdot t_M(K) \cdot (v_M(K) + t_M(K)).$$

Как правило,

$$t_M(K) \geq v_M(K).$$

Поэтому из теоремы 1.3 вытекает, что *отказ от использования многих головок приводит не более чем к квадратному растяжению времени вычислений*. Значение этого следствия определяется тем, что многоголовочная МТ с несколькими управляющими устройствами является естественной моделью для *параллельных вычислений*.

Многоленточная МТ с k рабочими не ограниченными в обе стороны лентами изображена на рис. 1.4. Эта МТ, как и предыдущая, также являет-

ся моделью для исследования *параллельных вычислений*. Входная (соответственно, выходная) лента предназначена только для считывания *входа* (соответственно, только для записи *выхода*), причем после считывания (соответственно, записи) очередного символа связанная с ней управляющая головка может сдвинуться только на клетку вправо. В 1965 году *J. Hartmanis* и *R.E. Stearns* доказали следующую теорему.

Теорема 1.4. Для любой одноленточной k -головочной МТ M существует такая эквивалентная ей многоленточная МТ M_1 , что

$$t_{M_1}(n) \leq c(M) \cdot t_M(n) \quad (n \in \mathbf{N}),$$

где $c(M)$ – константа, зависящая от МТ M .

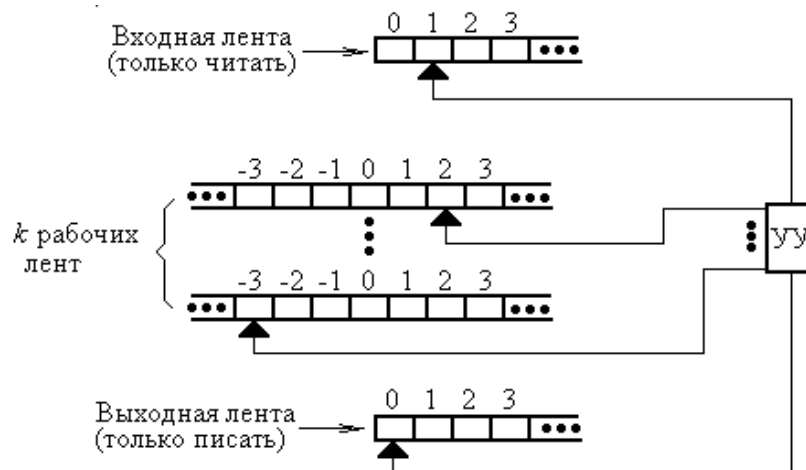


Рис. 1.4. Многоленточная МТ с k рабочими не ограниченными в обе стороны лентами.

Обычная n -мерная МТ $M = (Q, A, I, \delta, \Lambda, q_{in}, q_{fin})$ имеет одну ленту, являющуюся решеткой либо типа \mathbf{Z}^n , либо типа \mathbf{Z}_+^n (случай, когда $n = 2$, а лента имеет тип \mathbf{Z}^2 , изображен на рис. 1.5). Функция переходов δ такой МТ – это (возможно, частичное) отображение множества $Q \times A$ в множество $A \times Q \times \mathbf{V}^n$, где

$$\mathbf{V} = \{(\alpha_1, \dots, \alpha_n) \mid \alpha_j \in \{-1, 0, 1\}, \sum_{j=1}^n |\alpha_j| \leq 1\},$$

причем $\alpha_j = 1$ (соответственно, $\alpha_j = -1$) означает, что сдвиг головки происходит на одну клетку по j -й координате в положительном (соответственно, в отрицательном) направлении, а $\alpha_j = 0$ означает отсутствие сдвига головки по j -й координате.

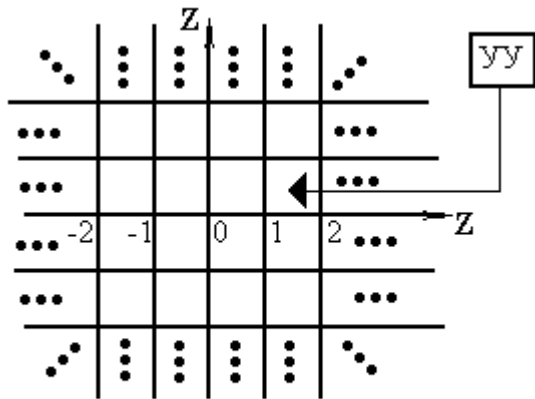


Рис. 1.5. Двумерная МТ с одной лентой типа Z^2 .

Если $n = 2$ (т.е. для двумерной МТ) принято функцию переходов δ представлять как отображение множества $Q \times A$ в множество $A \times Q \times \{Bв, Bн, Л, П, Нм\}$, где $Bв$, $Bн$, $Л$ и $П$ – сдвиг головки на одну клетку, соответственно, вверх, вниз, влево и вправо, а $Нм$ – головка остается на месте. В 1967 году Б.А. Трахтенброт доказал следующую теорему

Теорема 1.5. Для любой n -мерной МТ M существует такая эквивалентная ей многоленточная МТ M_1 с $(n + 1)$ -й рабочими не ограниченными в обе стороны лентами, что

$$t_{M_1}(n) \leq (t_M(n) + 1)(2n + t_M(n) + 2).$$

Так как, как правило, $t_M(n) \geq n$, то отказ от использования многомерных лент приводит не более чем к квадратному растяжению времени вычислений. Подчеркнем, что наиболее общей формальной моделью для *параллельных вычислений* (в рамках классического подхода к теории алгоритмов) является многоголовочная многоленточная МТ, возможно, с многомерными лентами.

В заключение рассмотрим кратко классификацию вычислений, осуществляемых МТ в зависимости от сложности процесса переработки конфигураций. Следуя традиции, в качестве базовой модели выберем обычную одноленточную МТ с одной не ограниченной в обе стороны лентой.

По аналогии с сигнализирующими времени и емкости определяются *сигнализирующие колебаний* $\omega_M(K)$ и *режима* $r_M(K)$. А именно, $\omega_M(K)$ и $r_M(K)$ равны, соответственно, числу колебаний (т.е. изменений направления головки) и максимальному числу прохождений головки через край клетки, если МТ M применима к конфигурации K и не определены в противном случае. Функции $\omega_M(n)$ и $r_M(n)$ определяются аналогично функциям $t_M(n)$ и $v_M(n)$. Говорят, что МТ M вычисляет значения функции f :

- 1) с *логарифмическим замедлением*, если

$$T_M(n) = O(n \cdot \log n) \quad (n \rightarrow \infty);$$

- 2) с *сублогарифмическим замедлением*, если

$$T_M(n) = o(n \cdot \log n) \quad (n \rightarrow \infty);$$

3) с ограниченным замедлением, если

$$T_M(n) = O(n) \quad (n \rightarrow \infty);$$

4) с ограниченным растяжением, если

$$V_M(n) = O(n) \quad (n \rightarrow \infty);$$

5) с ограниченным удлинением, если

$$v_M(n) \leq n + c,$$

где c – константа;

6) с ограниченным режимом, если

$$R_M(n) = O(1) \quad (n \rightarrow \infty),$$

где $R_M(n)$ – асимптотическая оценка $r_M(n)$.

Известно, что существуют следующие соотношения между перечисленными выше типами вычислений (см., напр. [11]).

Теорема 1.6. Пусть обычная одноленточная МТ M с одной не ограниченной в обе стороны лентой вычисляет значения функции f . Тогда:

- 1) если $R_M(n) = O(1) \quad (n \rightarrow \infty)$, то $v_M(n) \leq n + c$;
- 2) если $R_M(n) = O(1) \quad (n \rightarrow \infty)$, то существует такая МТ \tilde{M} , вычисляющая значения функции f , что $\omega_{\tilde{M}}(n) = 1$ для всех $n \in \mathbf{N}$;
- 3) если $R_M(n) = O(1) \quad (n \rightarrow \infty)$, то существует такая МТ \hat{M} , вычисляющая значения функции f , что $T_{\hat{M}}(n) = O(n) \quad (n \rightarrow \infty)$;
- 4) если $T_M(n) = O(n \cdot \log n) \quad (n \rightarrow \infty)$, то $V_M(n) = O(n) \quad (n \rightarrow \infty)$;
- 5) если $T_M(n) = o(n \cdot \log n) \quad (n \rightarrow \infty)$, то $R_M(n) = O(1) \quad (n \rightarrow \infty)$;
- 6) если $T_M(n) = O(n) \quad (n \rightarrow \infty)$, то $R_M(n) = O(1) \quad (n \rightarrow \infty)$;
- 7) если $T_M(n) = o(n \cdot \log n) \quad (n \rightarrow \infty)$, то $T_M(n) = O(n) \quad (n \rightarrow \infty)$ и $v_M(n) \leq n + c$.

Из 7-го утверждения теоремы 1.6 вытекает, что не существует МТ M , которая вычисляет значения данной функции с сигнализирующей времени, удовлетворяющей неравенствам

$$c_1 n \prec t_M(n) \prec c_2 n \cdot \log n \quad (n \in \mathbf{N}),$$

где c_1 и c_2 – константы. Верхняя и нижняя граница в этих неравенствах играют следующую роль (см., напр. [11]). Во-первых, любое рекурсивно-перечислимое множество может быть перечислено с логарифмическим замедлением. Во-вторых, класс множеств, распознаваемых с ограниченным режимом (и, следовательно, в силу 3-го утверждения теоремы 1.6, с линейным временем), совпадает с классом множеств, перечислимых с ограниченным режимом. Этот класс множеств Ω может быть охарактеризован следующим образом. *Настроенным конечным автоматом* называется система

$$M = (Q, X, \delta, q_{in}, Q_{fin}),$$

где Q – конечное множество состояний, X – конечный входной алфавит, $\delta : Q \times X \rightarrow Q$ – функция переходов, $q_{in} \in Q$ – начальное состояние и $Q_{fin} \subseteq Q$ – множество финальных состояний. Функция δ расширяется до функции $\tilde{\delta} : Q \times X^* \rightarrow Q$ с помощью равенств

$$\tilde{\delta}(q, \Lambda) \stackrel{опр}{=} q,$$

$$\tilde{\delta}(q, px) \stackrel{опр}{=} \delta(\tilde{\delta}(q, p), x)$$

для всех $q \in Q$, $p \in X^*$ и $x \in X$. Язык, представимый настроенным конечным автоматом

$$M = (Q, X, \delta, q_{in}, Q_{fin})$$

определяется равенством

$$L_M \stackrel{опр}{=} \{p \in X^* \mid \tilde{\delta}(q_{in}, p) \in Q_{fin}\}.$$

Указанный выше класс множеств Ω , распознаваемых с ограниченным режимом, совпадает с классом языков, представимых настроенными конечными автоматами.

1.2. Графы.

Граф определяется как упорядоченная пара

$$G = (V, E),$$

где V – множество вершин, а E ($E \subseteq V^{(2)}$) – множество ребер ($V^{(2)}$ – множество всех неупорядоченных пар элементов множества V). Если

$e = \{v_1, v_2\} \in E$, то вершины v_1 и v_2 – смежные, v_1 и v_2 инцидентны ребру e (v_1 и v_2 называются, также, концами ребра e), а ребро e инцидентно вершинам v_1 и v_2 . Ребра e_1 и e_2 ($e_1, e_2 \in E$) – смежные, если $|e_1 \cap e_2| = 1$, т.е. если они имеют (только один) общий конец.

В дальнейшем, если не оговорено противное, будем рассматривать только конечные графы, т.е. будем считать, что $|V| < \infty$. Если $|V| = n$ и $|E| = m$, то граф G имеет порядок $|G| = n$ и размер $e(G) = m$. Говорят, также, что G – (n, m) -граф. Для (n, m) -графа истинным является неравенство

$$0 \leq m \leq \binom{n}{2}.$$

Основными способами задания (n, m) -графа $G = (V, E)$ являются следующие:

1. *Перечисление* в явном виде элементов множеств V и E .
2. *Графическое изображение*, т.е. вершины изображаются кружками, а ребра – линиями, соединяющими пары кружков (в соответствии с этой интерпретацией иногда говорят, что ребро e соединят вершины v_1 и v_2 , если v_1 и v_2 инцидентны ребру e).
3. *Матрица смежности вершин*, т.е. булева матрица порядка $n \times n$, строки и столбцы которой занумерованы элементами множества V , а элемент матрицы, расположенный на пересечении строки $v_1 \in V$ и столбца $v_2 \in V$ равен 1 тогда и только тогда, когда v_1 и v_2 – смежные вершины.
4. *Матрица смежности ребер*, т.е. булева матрица порядка $m \times m$, строки и столбцы которой занумерованы элементами множества E , а элемент матрицы, расположенный на пересечении строки $e_1 \in E$ и столбца $e_2 \in E$ равен 1 тогда и только тогда, когда e_1 и e_2 – смежные ребра.
5. *Матрица инциденций*, т.е. булева матрица порядка $n \times m$, строки которой занумерованы элементами множества V , а столбцы – элементами множества E , причем элемент матрицы, расположенный на пересечении строки $v \in V$ и столбца $e \in E$ равен 1 тогда и только тогда, когда вершина v инцидентна ребру e .
6. *Списки смежности вершин*, т.е. совокупность n списков, занумерованных элементами множества V , причем список, занумерованный элементом $v \in V$, состоит из всех вершин, смежных с вершиной v .
7. *Списки смежности ребер*, т.е. совокупность m списков, занумерованных элементами множества E , причем список, занумерованный элементом $e \in E$, состоит из всех ребер, смежных с ребром e .

Пример 1.1. Граф $G_1 = (V, E)$, задан перечислением множеств вершин и ребер:

$$V = \{v_1, v_2, v_3, v_4, v_5\},$$

$$E = \{e_1 = \{v_1, v_3\}, e_2 = \{v_3, v_5\}, e_3 = \{v_3, v_4\}, e_4 = \{v_4, v_5\}, e_5 = \{v_2, v_5\}, e_6 = \{v_2, v_4\}\}.$$

Его графическое изображение показано на рис.1.6.

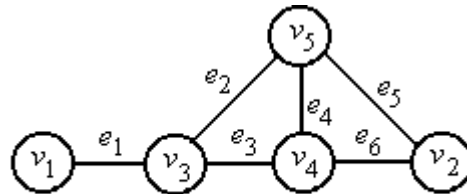


Рис. 1.6. Граф G_1 .

Матрицы смежности вершин и ребер, соответственно, \mathbf{M}_1 и \mathbf{M}_2 , а также матрица инциденций \mathbf{M}_3 графа $G_1 = (V, E)$ имеют следующий вид:

$$\mathbf{M}_1 = \begin{array}{c|ccccc} & v_1 & v_2 & v_3 & v_4 & v_5 \\ \hline v_1 & 0 & 0 & 1 & 0 & 0 \\ v_2 & 0 & 0 & 0 & 1 & 1 \\ v_3 & 1 & 0 & 0 & 1 & 1 \\ v_4 & 0 & 1 & 1 & 0 & 1 \\ v_5 & 0 & 1 & 1 & 1 & 0 \end{array},$$

$$\mathbf{M}_2 = \begin{array}{c|cccccc} & e_1 & e_2 & e_3 & e_4 & e_5 & e_6 \\ \hline e_1 & 0 & 1 & 1 & 0 & 0 & 0 \\ e_2 & 1 & 0 & 1 & 1 & 1 & 0 \\ e_3 & 1 & 1 & 0 & 1 & 0 & 1 \\ e_4 & 0 & 1 & 1 & 0 & 1 & 1 \\ e_5 & 0 & 1 & 0 & 1 & 0 & 1 \\ e_6 & 0 & 0 & 1 & 1 & 1 & 0 \end{array},$$

$$\mathbf{M}_3 = \begin{array}{c|cccccc} & e_1 & e_2 & e_3 & e_4 & e_5 & e_6 \\ \hline v_1 & 1 & 0 & 0 & 0 & 0 & 0 \\ v_2 & 0 & 0 & 0 & 0 & 1 & 1 \\ v_3 & 1 & 1 & 1 & 0 & 0 & 0 \\ v_4 & 0 & 0 & 1 & 1 & 0 & 1 \\ v_5 & 0 & 1 & 0 & 1 & 1 & 0 \end{array}.$$

Списки смежности, определяющие граф G_1 , показаны на рис. 1.7.

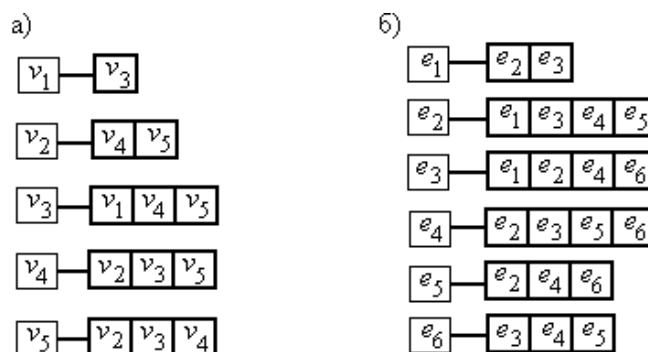


Рис. 1.7. Представление графа G_1 списками смежности: а) вершин; б) ребер.

При равномерном весе емкостная сложность задания (n, m) -графа перечислением равна $O(n + m)$ ($n, m \rightarrow \infty$), матрицами смежности вершин и ребер равна, соответственно, $O(n^2)$ ($n \rightarrow \infty$) и $O(m^2)$ ($n, m \rightarrow \infty$), матрицей инциденций — $O(nm)$ ($n, m \rightarrow \infty$), списками смежности вершин и ребер равна, соответственно, $O(\max\{n, m\})$ ($n, m \rightarrow \infty$) и $O(m \cdot \max\{1, \min\{n, m\}\})$ ($n, m \rightarrow \infty$).

Графы $G_1 = (V_1, E_1)$ и $G_2 = (V_2, E_2)$ равны (обозначается $G_1 = G_2$) тогда и только тогда, когда $V_1 = V_2$ и $E_1 = E_2$. Граф $G_1 = (V_1, E_1)$ — подграф графа $G = (V, E)$, если $V_1 \subseteq V$ и $E_1 \subseteq E$. Подграф $G_1 = (V_1, E_1)$ графа $G = (V, E)$ называется:

- 1) индуцированным множеством вершин V_1 , если $E_1 = E \cap V_1^{(2)}$;
- 2) остовным, если $V_1 = V$.

Отметим, что число остовных подграфов графа $G = (V, E)$ равно $2^{|E|}$.

Простейшими операциями над графом $G = (V, E)$ являются следующие:

- 1) $G - V_1$ ($V_1 \subseteq V$) — удаление вершин, принадлежащих множеству V_1 и инцидентных им ребер;
- 2) $G - E_1$ ($E_1 \subseteq E$) — удаление ребер, принадлежащих множеству E_1 ;
- 3) $G + E_1$ ($E_1 \subseteq V^{(2)} \setminus E$) — добавление ребер, принадлежащих E_1 ;
- 4) $\bar{G} = (V, V^{(2)} \setminus E)$ — дополнение графа G .

Для краткости ребро $e = \{v_1, v_2\} \in E$ обозначается через v_1v_2 (при этом записи v_1v_2 и v_2v_1 не различаются). Операции удаления вершины $v \in V^{(2)}$ и ребра $v_1v_2 \in E$ (соответственно, добавления ребра $v_1v_2 \in V^{(2)} \setminus E$) записываются в виде $G - v$ и $G - v_1v_2$ (соответственно $G + v_1v_2$).

В графе $G = (V, E)$ множества вершин

$$\Gamma(v) = \{v_1 \in V \mid vv_1 \in E\}$$

и

$$\Gamma(v) \cup \{v\} \quad (v \in V)$$

называются, соответственно, *открытой* и *замкнутой окрестностью* вершины v . Таким образом, задание графа списками смежности – это перечисление замкнутых окрестностей всех его вершин. Число

$$d(v) = |\Gamma(v)| \quad (v \in V)$$

называется *степенью вершины* v . Как правило (хотя это не является обязательным требованием), вершины графа нумеруются в порядке не убывания их степеней, т.е. если $G = (V, E) = (n, m)$ -граф, то предполагается, что

$$d(v_1) \leq \dots \leq d(v_n)$$

(отметим, что в примере 1.1 выбрана именно такая нумерация вершин графа G_1). В 1736 году *L. Euler* установил следующее свойство последовательности степеней вершин графа.

Теорема 1.7. (*The handshaking lemma*). Сумма степеней всех вершин графа $G = (V, E)$ равна удвоенному числу его ребер, т.е.

$$\sum_{v \in V} d(v) = 2e(G).$$

Из теоремы 1.7 вытекает, что для любого графа $G = (V, E)$

$$\delta(G) = \min_{v \in V}^{oпр} d(v) \leq \lfloor 2e(G) \cdot |V|^{-1} \rfloor,$$

$$\Delta(G) = \max_{v \in V}^{oпр} d(v) \geq \lceil 2e(G) \cdot |V|^{-1} \rceil.$$

Если $\delta(G) = \Delta(G) = k$, то граф G – *k-регулярный*. Экстремальными случаями регулярного графа являются:

- 1) *полный граф* K_n порядка n , т.е. $(n, \binom{n}{2})$ -граф ($n \in \mathbf{N}$) (в котором любые две вершины – смежные);
- 2) *пустой граф* \bar{K}_n порядка n (в котором вообще нет ребер).

В частности, полный граф K_3 порядка 3 называется *треугольником*. *W. Mantel* в 1907 году доказал следующее достаточное условие, при котором треугольник является подграфом данного графа.

Теорема 1.8. Граф $G = (V, E)$ содержит подграф, являющийся треугольником, если

$$e(G) > \lfloor 0.25 \cdot |G|^2 \rfloor.$$

В 1941 году *P. Turán* показал, что установленная в теореме 1.8 оценка является точной нижней границей.

Подграф

$$P = (V(P), E(P))$$

графа $G = (V, E)$, где

$$V(P) = \{v_0, \dots, v_l\}$$

и

$$E(P) = \{v_0v_1, v_1v_2, \dots, v_{l-1}v_l\} \quad (v_{i-1}v_i \in E \quad (i = 1, \dots, l)),$$

называется *путем длины l (в графе G)*, соединяющим вершины v_0 и v_l . Говорят также, что P – $(v_0 - v_l)$ -*путь* (в этом случае вершины v_0 и v_l выделяется в качестве, соответственно, *начальной* и *финальной* вершин пути). В графе $G = (V, E)$ $(v_0 - v_l)$ -пути $P_1 = (V(P_1), E(P_1))$ и $P_2 = (V(P_2), E(P_2))$ называются *независимыми*, если $V(P_1) \cap V(P_2) = \{v_0, v_l\}$.

Понятие ‘*путь в графе*’ $G = (V, E)$ дает возможность определить *метрику* (т.е. *расстояние*) на множестве V вершин следующим образом:

- 1) $d(v, v) = 0$ для всех $v \in V$;
- 2) $d(v_1, v_2)$ ($v_1 \neq v_2$) равно наименьшей длине $(v_1 - v_2)$ -пути, если такой путь существует;
- 3) $d(v_1, v_2) = \infty$ ($v_1 \neq v_2$), если $(v_1 - v_2)$ -путь не существует.

Введенная метрика, в свою очередь, дает возможность определить *диаметр* и *радиус* графа $G = (V, E)$ с помощью, соответственно, равенств

$$diam G = \max_{v_1, v_2 \in V}^{опр} d(v_1, v_2),$$

$$rad G = \min_{v_1 \in V}^{опр} \max_{v_2 \in V} d(v_1, v_2).$$

Величина $\max_{v_2 \in V} d(v_1, v_2)$ называется *эксцентриситетом* вершины v_1 , а величина

$$\text{cntr } G = \{v_1 \in V \mid \max_{v_2 \in V} d(v_1, v_2) = \text{rad } G\}$$

называется *центром* графа $G = (V, E)$.

Граф $G = (V, E)$ – *связный*, если для любых двух его вершин v_1 и v_2 существует $(v_1 - v_2)$ -путь. Максимальные связные подграфы графа $G = (V, E)$ называются его *компонентами связности*. Отметим, что любая такая вершина v графа G , что $d(v) = 0$ (она называется *изолированной*) является его компонентой связности. Операции удаления вершин или ребер графа могут изменить количество компонент связности. В этом контексте выделяются следующие вершины и ребра:

1) вершина v графа G называется *точкой сочленения*, если число компонент связности графа $G - v$ больше числа компонент связности графа G ;

2) ребро $v_1 v_2$ графа G называется *мостом*, если число компонент связности графа $G - v_1 v_2$ больше числа компонент связности графа G .

В графе $G = (V, E)$ *прогулка* (a walk) (используется, также, термин *маршрут*) длины l – это такая чередующаяся последовательность вершин и ребер

$$v_0, e_1, v_1, e_2, \dots, e_l, v_l,$$

что $e_i = v_{i-1} v_i$ ($i = 1, \dots, l$). Такое определение предполагает, что зафиксирована *ориентация* от начальной вершины v_0 к финальной вершине v_l , что подчеркивается использованием термина $(v_0 - v_l)$ -прогулка. *След* (a trail) (используется, также, термин *цепь*) – это прогулка, в которой все ребра e_1, e_2, \dots, e_l попарно различны. Отметим, что *путь* – это прогулка (или след) с попарно различными вершинами. След

$$v_0, e_1, v_1, e_2, \dots, e_l, v_l$$

называется *контуром* (a circuit), если $v_0 = v_l$. Для обозначения контура

$$v_0, e_1, v_1, e_2, \dots, e_l, v_0$$

используется запись

$$v_0 v_1 \dots v_{l-1}$$

(подчеркнем, что контур $v_0v_1 \dots v_{l-1}$ содержит ребро $v_{l-1}v_0$). Для контура начальная вершина не выделяется и отсутствует ориентация, т.е. записи

$$v_0v_1 \dots v_{l-1}, v_1 \dots v_{l-1}v_0, \dots, v_{l-1}v_0 \dots v_{l-2}, v_{l-1}v_{l-2} \dots v_0, v_{l-2} \dots v_0v_{l-1}, \dots, v_0v_{l-1} \dots v_1$$

эквивалентны (они определяют один и тот же контур). Контур $v_0v_1 \dots v_{l-1}$ называется *циклом длины l* , если $l \geq 3$ и вершины v_0, v_1, \dots, v_{l-1} – попарно различные. Для цикла (как и для контура) отсутствует ориентация.

Для связных графов выделяются следующие важные специальные случаи введенных выше понятий *контур*, *прогулка*, *путь*, *след* и *цикл*. Путь или цикл – *гамильтонов*, если он содержит все вершины графа. Контур, след или прогулка – *эйлеров*, если он содержит все ребра графа. Отметим, что решение ряда модельных задач дискретной математики, таких, как задачи *коммивояжера* и *китайского почтальона* (см., напр. [5,7,12,14]) основаны именно на поиске специальных гамильтонового цикла и эйлеровой прогулки. В 1736 году *L. Euler* установил следующий критерий существования в графе эйлеровых контура и следа.

Теорема 1.9. Связный граф содержит:

- 1) эйлеров контур тогда и только тогда, когда степень каждой его вершины – четная;
- 2) эйлеров (v_1, v_2) -след ($v_1 \neq v_2$) тогда и только тогда, когда v_1 и v_2 – единственные вершины графа, имеющие нечетную степень.

Понятие ‘*цикл в графе*’ имеет многочисленные применения при исследовании структуры графа. Рассмотрим их кратко.

Подмножество вершин (соответственно, подмножество ребер) графа – *независимое*, если никакие два его элемента не являются смежными. Мощность максимального по числу элементов независимого подмножества вершин (соответственно, ребер) называется *вершинным* (соответственно, *реберным*) *числом независимости* графа G и обозначается $\beta_0(G)$ (соответственно, $\beta_1(G)$). Граф $G = (V, E)$ – *r -дольный*, если существует разбиение $\pi = \{V_1, \dots, V_r\}$ множества его вершин V на независимые подмножества вершин. В 1936 году *D. König* установил следующий критерий.

Теорема 1.10. Граф является *двудольным* тогда и только тогда, когда он не содержит циклов нечетной длины.

r -дольный граф $G = (V, E)$ (где $\pi = \{V_1, \dots, V_r\}$ – разбиение множества V на независимые подмножества) называется *полным* (обозначается $K_{|V_1|, \dots, |V_r|}$), если каждая вершина каждого множества V_i и каждая вершина каждого множества V_j ($j \neq i$) – смежные, т.е. если

$$E = \bigcup_{\substack{i,j=1 \\ i \neq j}}^r \{v_1 v_2 \mid v_1 \in V_i, v_2 \in V_j\}.$$

В 1912 году *O. Veblen* установил следующую связь между циклами в графе и разбиениями множества его ребер.

Теорема 1.11. Множество ребер графа $G = (V, E)$ может быть разбито на циклы тогда и только тогда, когда степень $d(v)$ каждой вершины $v \in V$ – четное число.

Этот результат, по своей сути, является важным специальным случаем следующей конструкции, устанавливающей тесную связь между графами и векторными пространствами над конечными полями. Пусть $G = (V, E)$ – (n, m) -граф и $E = \{e_1, \dots, e_m\}$. Представим каждый цикл C в графе G двоичным вектором

$$\mathbf{c} = (\alpha_1, \dots, \alpha_m),$$

где $\alpha_i = 1$ ($i = 1, \dots, m$) тогда и только тогда, когда ребро e_i принадлежит циклу C . Обозначим через

$$\mathbf{C} = \{\mathbf{c}_1, \dots, \mathbf{c}_k\}$$

множество двоичных векторов, представляющих множество всех циклов $\{C_1, \dots, C_k\}$ в графе G . Тогда множество

$$\mathbf{V}_c(G) = \begin{cases} \{\mathbf{0}\}, & \text{если } \mathbf{C} = \emptyset \\ \left\{ \sum_{i=1}^k \beta_i \mathbf{c}_i \mid \beta_i \in \{0,1\} (i = 1, \dots, k) \right\}, & \text{если } \mathbf{C} \neq \emptyset \end{cases}$$

где ‘+’ – операция покомпонентного сложения по mod 2 векторов, является подпространством векторного пространства $\mathbf{GF}^m(2)$. Подпространство $\mathbf{V}_c(G)$ называется *пространством циклов графа G* , а число

$$r_c(G) = \text{Dim } \mathbf{V}_c(G)$$

называется *циклическим рангом* (или *цикломатическим числом*) графа G . Известно, что (см., напр. [12]) для любого (n, m) -графа $G = (V, E)$ с k компонентами связности истинным является уравнение *Эйлера-Пуанкаре для графов*

$$n - m = k - r_c(G).$$

Циклы играют важную роль при *характеристике топологических свойств графа*. В 1936 году *D. König* показал, что любой (n, m) -граф $G = (V, E)$, можно изобразить (или, иными словами, *уложить*) без *самопересечений* (т.е. любая общая точка двух ребер – вершина графа) на *сфере с ручками*. Наименьшее число $\gamma(G)$ ручек, которые нужно добавить к сфере для такой укладки, называется *родом* графа G . Уложим граф G без самопересечений на сфере $S_{\gamma(G)}$, имеющей $\gamma(G)$ ручек. Обозначим через $F(G)$ множество связных областей, полученных в результате удаления из $S_{\gamma(G)}$ всех точек, принадлежащих ребрам графа G и всех точек, изображающих вершины графа G . Элементы множества $F(G)$ называются *гранями* (или *сторонами*) графа G . В 1736 году *L. Euler* установил следующий факт.

Теорема 1.12. Для связного (n, m) -графа $G = (V, E)$ истинным является равенство

$$n - m + |F(G)| = 2 - 2\gamma(G).$$

Число $n - m + |F(G)|$ называется *эйлеровой характеристикой* графа G . Из уравнения Эйлера-Пуанкаре и теоремы 1.12 вытекает, что для связного графа G истинным является равенство

$$r_c(G) = 2\gamma(G) + |F(G)| - 1.$$

Граф G – *планарный*, если $\gamma(G) = 0$, т.е. если G можно уложить без самопересечений на обычной сфере. Такие графы характеризуются тем, что именно их можно уложить без самопересечений на плоскости (такая укладка называется *плоским графом*). Для связного планарного графа G эйлерова характеристика равна 2, а циклический ранг равен

$$r_c(G) = |F(G)| - 1.$$

Для планарного графа $G = (V, E)$ известна более тонкая топологическая характеристика, чем свойство $\gamma(G) = 0$. Операция *разбиение ребра* состоит в замене ребра путем длины 2 (т.е. если $v_i v_j \in E$, то полагаем $V := V \cup \{v\}$ ($v \notin V$) и $E := E \setminus \{v_i v_j\} \cup \{v_i v, v v_j\}$). Графы G_1 и G_2 – *гомеоморфные*, если они могут быть получены из одного и того же графа G в результате конечных последовательностей операций разбиения ребер (возможно, с последующим изменением обозначений вершин). В 1930 году *K. Kuratowski* установил следующий критерий планарности графа.

Теорема 1.13. Граф G – планарный тогда и только тогда, когда он не содержит подграфа, гомеоморфного K_5 или $K_{3,3}$.

Граф называется *ациклическим* (или *лесом*), если он не содержит циклов. Известно, что любая последовательность натуральных чисел d_1, \dots, d_n , удовлетворяющая равенству

$$\sum_{i=1}^n d_i = 2(n - k) \quad (1 \leq k \leq n - 1),$$

является последовательностью степеней вершин некоторого леса с k компонентами связности. *Деревом* называется связный ациклический граф. Деревья играют важную роль во многих проблемах исследования связных графов. По этой причине выделяются *остовные деревья* (или, более кратко *остовы*) связных графов, т.е. остовные подграфы, являющиеся деревьями.

Пример 1.2. Число остовных подграфов графа G_1 из примера 1.1, равно $2^6 (= 64)$. Среди них остовными деревьями являются 8 подграфов (каждый с множеством вершин $V = \{v_1, v_2, v_3, v_4, v_5\}$), определяемых следующими множествами ребер

$$E_1 = \{e_1, e_2, e_3, e_5\}, E_2 = \{e_1, e_2, e_3, e_6\}, E_3 = \{e_1, e_2, e_4, e_5\}, E_4 = \{e_1, e_2, e_4, e_6\},$$

$$E_5 = \{e_1, e_2, e_5, e_6\}, E_6 = \{e_1, e_3, e_4, e_5\}, E_7 = \{e_1, e_3, e_4, e_6\}, E_8 = \{e_1, e_3, e_5, e_6\}.$$

Любое остовное дерево $T = (V, E_T)$ связного (n, m) -графа $G = (V, E)$ содержит минимальное число ребер, связывающих все вершины графа G . Ясно, что если $E \setminus E_T \neq \emptyset$, то граф $T + v$ ($v \in E \setminus E_T$) содержит единственный цикл, причем, циклы, получаемые при добавлении к T различных ребер $v \in E \setminus E_T$ – линейно независимые (так как каждый цикл содержит ребро, не принадлежащее никакому другому из этих циклов). Число этих циклов равно

$$m - (n - 1) = m - n + 1 = r_c(G).$$

Следовательно, указанные выше циклы образуют базис векторного пространства $V_c(G)$.

Иногда возникает необходимость *размечать* вершины и/или ребра графа элементами некоторого множества. Формально, *разметка вершин* (соответственно, *ребер*) графа $G = (V, E)$ элементами множества Σ (или, более кратко, Σ -разметка) – это отображение $f : V \rightarrow \Sigma$ (соответственно, $f : E \rightarrow \Sigma$) (возможно удовлетворяющее некоторым дополнительным условиям). Среди Σ -разметок вершин (соответственно, ребер) графа $G = (V, E)$ выделяют *раскраски*, т.е. Σ -разметки, у которых образы любых двух смежных вершин (соответственно, ребер) – различные. Мощность минимального (по числу элементов) множества Σ , для которого существу-

ет Σ -раскраска вершин (соответственно, ребер) графа G называется *вершинно-хроматическим* (соответственно, *реберно-хроматическим*) числом графа G и обозначается через $\chi(G)$ (соответственно, через $\chi'(G)$). От раскраски ребер легко перейти к раскраске вершин, если заменить граф $G = (V, E)$ его *реберным графом* $L(G) = (E, \tilde{V})$, где

$$\tilde{V} = \{\{e_1, e_2\} \in E^{(2)} \mid e_1 \text{ и } e_2 \text{ - смежные ребра графа } G\}.$$

Отметим, что если G – (n, m) -граф, то $L(G)$ – это $(m, 0.5 \cdot \sum_{i=1}^n d_i^2 - m)$ -граф, где d_1, \dots, d_n – последовательность степеней вершин графа G . Известен ряд оценок для вершинно-хроматического числа графа, среди которых отметим оценки

$$\begin{cases} 2\sqrt{n} \leq \chi(G) + \chi(\bar{G}) \leq n + 1 \\ n \leq \chi(G) \cdot \chi(\bar{G}) \leq (0.5 \cdot (n + 1))^2 \end{cases}$$

которые установили в 1956 году *E.A. Nordhaus* и *J.W. Gaddum*, и оценки

$$n \cdot (\beta_0(G))^{-1} \leq \chi(G) \leq n - \beta_0(G) + 1$$

нижнюю из которых, по видимому, установил в 1958 году *C. Berge*, а верхнюю – в 1970 году *F. Harary* и *S. Hedetniemi*.

До сих пор использовалось определение графа $G = (V, E)$, которое заведомо предполагает, что вершины *занумерованы* (т.е. взаимно однозначно *размечены*) элементами множества V . Переход к графам с *не размеченными вершинами* естественно осуществляется следующим образом.

Графы $G_1 = (V_1, E_1)$ и $G_2 = (V_2, E_2)$ называются *изоморфными*, если существует такая биекция

$$\varphi: V_1 \rightarrow V_2,$$

что

$$v_1 v_2 \in E_1 \Leftrightarrow \varphi(v_1) \varphi(v_2) \in E_2.$$

Переход к графам с не размеченными вершинами, по своей сути, означает переход к классам эквивалентности, определяемой отношением изоморфизма графов.

Часто приходится рассматривать конструкции, не укладывающиеся в приведенное в начале настоящего пункта определение графа $G = (V, E)$ (иногда графы, удовлетворяющие приведенному в начале настоящего

пункта определению, называют *простыми* или *обычными* графами). Эти отличия могут состоять в следующем.

Во-первых, допускаются *петли* (т.е. ребра вида $\{v, v\}$, соединяющие вершину с собой) и *кратные ребра* (т.е. несколько ребер, соединяющих данную пару вершин). Если допускаются только петли, то конструкция называется *графом с петлями*. В графе с петлями считается, что вклад каждой петли в степень вершины равен 2. Если же допускаются и петли, и кратные дуги, то конструкция называется *мультиграфом*. В матрице смежности вершин мультиграфа (не являющейся булевой) элемент, расположенный на пересечении строки $v \in V$ и столбца $v' \in V$ равен количеству ребер, соединяющих эти вершины.

Во-вторых, иногда множество ребер определяют как подмножество множества

$$V \times V \setminus \{(v, v) \mid v \in V\}.$$

Элементы этого множества называются *дугами*. Для краткости записи дуга (v_1, v_2) обозначается через $\overrightarrow{v_1 v_2}$ (v_1 – начало, а v_2 – конец дуги). Такая конструкция – *направленный граф (directed graph)*.

Основные отличия введенных выше понятий для направленного графа состоят в следующем. На пересечении строки $v \in V$ и столбца $v' \in V$ матрицы смежности вершин расположен элемент 1 тогда и только тогда, когда $\overrightarrow{vv'} \in E$. На пересечении строки $\vec{e} \in E$ и столбца $\vec{e}' \in V$ матрицы смежности дуг расположен элемент 1 тогда и только тогда, когда конец дуги \vec{e} совпадает с началом дуги \vec{e}' . Матрица инцидентий не является булевой. В ней на пересечении строки $v \in V$ и столбца $\vec{e} \in E$ расположен элемент 1, если v начало дуги \vec{e} , элемент -1 , если v конец дуги \vec{e} и элемент 0 в остальных случаях (т.е. в каждом столбце матрицы инцидентий направленного графа существует ровно два отличных от нуля элементов, один из которых равен 1, а другой -1). В направленном графе *полустепени исхода* и *захода* для вершины v определяются равенствами

$$d^+(v) \stackrel{оп}{=} |\{v_1 \in V \mid \overrightarrow{vv_1} \in E\}|,$$

$$d^-(v) \stackrel{оп}{=} |\{v_1 \in V \mid \overrightarrow{v_1 v} \in E\}|,$$

а *прогулка* – это такая чередующаяся последовательность вершин и дуг $v_0, e_1, v_1, e_2, \dots, e_l, v_l$, что $e_i = \overrightarrow{v_{i-1} v_i}$ ($i = 1, \dots, l$). Контур, путь, след и цикл определяются по аналогии с определениями для обычного графа, как специальные случаи прогулки. В направленном графе могут допускаться *петли* (т.е. упорядоченные пары вида \overrightarrow{vv}) и *кратные дуги* (т.е. несколько дуг

имеющих одно и то же начало и конец). Если допускаются только петли, то конструкция называется *направленным графом с петлями*. Если же допускаются и петли, и кратные дуги, то конструкция называется *направленным мультиграфом*.

В-третьих, иногда в обычном графе $G = (V, E)$ удобно (или необходимо) выбрать для каждого ребра $\{v_1, v_2\}$ *ориентацию*, т.е. заменить его либо дугой $\overrightarrow{v_1 v_2}$, либо дугой $\overrightarrow{v_2 v_1}$. Конструкция, полученная в результате ориентации всех ребер обычного графа называется *ориентированным графом*, или, более коротко, *орграфом*. Ясно, что орграф – это такой специальный случай направленного графа, что существует не более одной дуги, соединяющей любую пару вершин. В 1954 году *Н.М. Трент* установил следующий способ подсчета числа остовных деревьев в графах, являющийся иллюстрацией эффективного использования перехода от обычного графа к орграфу

Теорема 1.14. Пусть \mathbf{M} – матрица инциденций орграфа, полученного в результате какой-либо ориентации ребер обычного графа G , \mathbf{M}_0 – матрица, полученная в результате удаления какой-нибудь одной строки матрицы \mathbf{M} , а \mathbf{M}_0^T – матрица, транспонированная к \mathbf{M}_0 . Число остовных деревьев графа G равно определителю $|\mathbf{M}_0 \cdot \mathbf{M}_0^T|$.

Пример 1.3. Преобразуем граф G_1 из примера 1.1 в орграф, выбрав ориентацию его ребер от вершины с меньшим индексом к вершине с большим индексом. Матрица инциденций полученного орграфа имеет следующий вид

$$\mathbf{M} = \begin{array}{c|cccccc} & e_1 & e_2 & e_3 & e_4 & e_5 & e_6 \\ \hline v_1 & 1 & 0 & 0 & 0 & 0 & 0 \\ v_2 & 0 & 0 & 0 & 0 & 1 & 1 \\ v_3 & -1 & 1 & 1 & 0 & 0 & 0 \\ v_4 & 0 & 0 & -1 & 1 & 0 & -1 \\ v_5 & 0 & -1 & 0 & -1 & -1 & 0 \end{array} .$$

Обозначим через \mathbf{M}_0 матрицу, полученную в результате удаления третьей строки в матрице \mathbf{M} . Тогда

$$\mathbf{M}_0 \cdot \mathbf{M}_0^T = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 2 & -1 & -1 \\ 0 & -1 & 3 & -1 \\ 0 & -1 & -1 & 3 \end{pmatrix} .$$

Следовательно, число остовных деревьев графа G_1 из примера 1.1 равно

$$|\mathbf{M}_0 \cdot \mathbf{M}_0^T| = 8,$$

что согласуется с примером 1.2.

В заключение отметим, что если оргграф G_1 получен из обычного графа G , то

$$d_G(v) = d_{G_1}^+(v) + d_{G_1}^-(v)$$

для любой вершины v (нижний индекс при букве d отмечает объект, к которому относится параметр).

1.3. Автоматы в лабиринтах.

Проблема поиска выхода из лабиринта привлекала к себе внимание человечества на протяжении веков и известна из глубокой древности (миф о Критском лабиринте). В 1951 году *C. Shannon* исследовал поиск конечным автоматом выхода из конечного лабиринта. Эти результаты заложили основы и стимулировали развитие *исследования поведения автоматов в лабиринтах* – научного направления, лежащего на стыке 3-х теорий, а именно: алгоритмов, графов и автоматов. Отметим, что в настоящее время одна из, по-видимому, наиболее мощных школ в мире, занимающаяся исследованием поведения автоматов в лабиринтах, состоит из учеников проф. *Кудрявцева В.Б.* (Москва, МГУ). Говоря неформально, *лабиринт* – это обычный связный граф с, возможно, размеченными вершинами и средствами, позволяющими указать в явном виде направление прохода каждого ребра, так как предмет исследования – анализ свойств тех или иных прогулок по графу. Формально понятие *лабиринт* определяется следующим образом (см., напр. [6]).

Пусть $G = (V, E)$ ($|V| \leq \aleph_0$) – такой связный граф, что при подходящем значении $n \in \mathbf{N}$ граф G можно так уложить без самопересечений в пространстве \mathbf{R}^n , что каждое ребро – это отрезок прямой в \mathbf{R}^n , параллельной некоторой координатной оси. Из этого предположения вытекает, что окрестность каждой вершины $v \in V$ – это *конечное множество*. Более того,

$$|\Gamma(v)| \leq n$$

для всех $v \in V$. В настоящем пункте под графом G понимается именно такая его укладка в \mathbf{R}^n . Предполагается, что задана Ω -разметка ($|\Omega| < \infty$, $\Lambda \in \Omega$) вершин графа G . В частности, если $\Omega = \{\Lambda\}$, то v ($v \in V$) – единственная отметка вершины графа.

При проходе ребра $v_i v_j \in E$ необходимо различать *в каком направлении происходит* этот *проход* (от вершины v к вершине v' или, наоборот, от вершины v' к вершине v), что осуществляется следующим образом. Фиксируется алфавит

$$\Sigma = \{\sigma_1^{(+)}, \dots, \sigma_n^{(+)}, \sigma_1^{(-)}, \dots, \sigma_n^{(-)}\},$$

где $\sigma_i^{(+)}$ (соответственно, $\sigma_i^{(-)}$) ($i = 1, \dots, n$) интерпретируется как утверждение: *движение осуществляется параллельно i -й координатной оси в положительном (соответственно, в отрицательном) направлении*. Каждое ребро $vv' \in E$ заменяется парой противоположно направленных дуг $\overrightarrow{vv'}$ и $\overleftarrow{v'v}$ (т.е. каждый отрезок $vv' \in E$ заменяется парой противоположно направленных векторов $\overrightarrow{vv'}$ и $\overleftarrow{v'v}$). Осуществляется Σ -разметка всех дуг в соответствии с правилом: дуга $\overrightarrow{vv'}$ отмечается символом $\sigma_i^{(+)}$, если вектор $\overrightarrow{vv'}$ сонаправлен с i -й координатной осью и символом $\sigma_i^{(-)}$, если вектор $\overrightarrow{vv'}$ направлен противоположно i -й координатной оси.

Прямоугольным n -мерным лабиринтом называется описанная выше укладка $L = (V, E)$ связного графа в \mathbf{R}^n с некоторой Ω -разметкой вершин и Σ -разметкой дуг. Если $n = 2$, то лабиринт – *плоский*. Множество $\mathbf{L}(n)$ всех прямоугольных n -мерных лабиринтов $L = (V, E)$ естественно разбивается на подмножества $\mathbf{L}_{<\aleph_0}(n)$ *конечных* (если $|V| < \aleph_0$) и *бесконечных* $\mathbf{L}_{=\aleph_0}(n)$ (если $|V| = \aleph_0$) лабиринтов. Выделяют следующие специальные классы лабиринтов, принадлежащих $\mathbf{L}(n)$. Класс $\mathbf{L}_y(n)$ *целочисленных n -мерных лабиринтов* состоит из всех таких лабиринтов $L = (V, E) \in \mathbf{L}(n)$, что $V \subseteq \mathbf{Z}^n$. Класс $\mathbf{L}_m(n)$ *мозаичных n -мерных лабиринтов* состоит из всех таких лабиринтов $L = (V, E) \in \mathbf{L}_y(n)$, что каждое ребро $vv' \in E$ – отрезок длины 1. Любой мозаичный n -мерный лабиринт является связным подграфом n -регулярного лабиринта $L_{\mathbf{Z}^n} = (\mathbf{Z}^n, E_{\mathbf{Z}^n})$, где

$$E_{\mathbf{Z}^n} = \{vv' \mid v, v' \in \mathbf{Z}^n \text{ и либо } \overrightarrow{vv'}, \text{ либо } \overleftarrow{v'v} - \text{орт}\}.$$

Класс $\mathbf{L}_u(n)$ *шахматных n -мерных лабиринтов* состоит из всех таких лабиринтов $L = (V, E) \in \mathbf{L}_m(n)$, что L – связный подграф лабиринта $L_{\mathbf{Z}^n}$, индуцированный множеством вершин V . Компоненты связности графа

$$L_{\mathbf{Z}^n} \setminus L = (\mathbf{Z}^n \setminus V, E_{\mathbf{Z}^n} \cap (\mathbf{Z}^n \setminus V) \times (\mathbf{Z}^n \setminus V))$$

называются дырами в мозаичном n -мерном лабиринте $L = (V, E) \in \mathbf{L}_m(n)$. Мозаичный n -мерный лабиринт $L \in \mathbf{L}_m(n)$ является $(r+1)$ -связным, если он имеет r конечных дыр.

Автоматом (Мили) называется система

$$M = (Q, X, Y, \delta, \lambda),$$

где Q , X и Y – непустые конечные множества состояний, входной и выходной алфавиты, а отображения

$$\delta : Q \times X \rightarrow Q$$

и

$$\lambda : Q \times X \rightarrow Y$$

называются, соответственно, функцией переходов и функцией выходов.

Говорят, что автомат M – допустимый для класса лабиринтов $\mathbf{L}(n)$, если

$$X = \Omega \times (\mathbf{B}(\Sigma) \setminus \{\emptyset\}),$$

где $\mathbf{B}(S)$ – булеан множества S ,

$$Y = \Sigma \cup \{h\},$$

а функция выходов λ удовлетворяет условию

$$\lambda(q, (\omega, \{\sigma_{i_1}, \dots, \sigma_{i_h}\})) \in \{\sigma_{i_1}, \dots, \sigma_{i_h}\} \cup \{h\}$$

для всех $q \in Q$, $\omega \in \Omega$ и $\{\sigma_{i_1}, \dots, \sigma_{i_h}\} \in \mathbf{B}(\Sigma) \setminus \{\emptyset\}$.

Такт работы допустимого автомата M в лабиринте $L \in \mathbf{L}(n)$ осуществляется следующим образом. Пусть в момент времени t ($t \in \mathbf{Z}_+$) автомат M в состоянии q находится в вершине v лабиринта L и $\Gamma(v) = \{v_{i_1}, \dots, v_{i_h}\}$.

Обозначим через $\sigma(\overrightarrow{vv_{i_j}})$ ($j = 1, \dots, h$) отметку дуги $\overrightarrow{vv_{i_j}}$. Если

$$\lambda(q, (\omega, \{\sigma(\overrightarrow{vv_{i_1}}), \dots, \sigma(\overrightarrow{vv_{i_h}})\})) = \sigma(\overrightarrow{vv_{i_j}}),$$

то к моменту времени $t+1$ автомат M перемещается в вершину v_{i_j} , а если

$$\lambda(q, (\omega, \{\sigma(\overrightarrow{vv_{i_1}}), \dots, \sigma(\overrightarrow{vv_{i_h}})\})) = h,$$

то M остается в вершине v . В обоих случаях к моменту времени $t+1$ автомат M переходит в состояние $\delta(q, (\omega, \{\sigma(\overrightarrow{vv_{i_1}}), \dots, \sigma(\overrightarrow{vv_{i_h}})\}))$.

Инициальный автомат – это упорядоченная пара

$$M_{q_{in}} = (M, q_{in}),$$

где $q_{in} \in Q$ – выделенное *начальное состояние*, а *инициальный лабиринт* – это упорядоченная пара

$$L_{v_{in}} = (L, v_{in}) \quad (L = (V, E) \in \mathbf{L}(n)),$$

где $v_{in} \in V$ – выделенная *начальная вершина*.

Если допустимый инициальный автомат $M_{q_0} = (M, q_0)$ помещен в инициальный лабиринт $L_{v_0} = (L, v_0)$ (т.е. в начальный момент времени M в состоянии q_0 находится в вершине v_0 лабиринта L), то в процессе своего функционирования он порождает бесконечную последовательность

$$\pi(M_{q_0}, L_{v_0}) = (q_0, v_0), (q_1, v_1), (q_2, v_2), \dots,$$

где v_{i+1} ($i \in \mathbf{Z}_+$) – вершина, в которую M , находясь в состоянии q_i , переходит из вершины v_i , а q_{i+1} – состояние, в которое переходит M при этом переходе по лабиринту. Последовательность $\pi(M_{q_0}, L_{v_0})$ называется *поведением инициального автомата M_{q_0} в инициальном лабиринте L_{v_0}* .

Выделим в последовательности $\pi(M_{q_0}, L_{v_0})$ вторую проекцию

$$pr_2 \pi(M_{q_0}, L_{v_0}) = v_0, v_1, v_2, \dots$$

Положим

$$Int(M_{q_0}, L_{v_0}) = \{v_i \mid i \in \mathbf{Z}_+\}.$$

Говорят, что инициальный автомат M_{q_0} *обходит* инициальный лабиринт L_{v_0} (в этом случае последовательность $pr_2 \pi(M_{q_0}, L_{v_0})$ – *обход*), если

$$Int(M_{q_0}, L_{v_0}) = V$$

и инициальный лабиринт L_{v_0} – *ловушка* для инициального автомата M_{q_0} , если

$$Int(M_{q_0}, L_{v_0}) \subset V.$$

Поведение допустимого инициального автомата в инициальном лабиринте является важным специальным случаем следующего, более общего, понятия. Пусть система

$$M = ((M_1)_{q_0^{(1)}}, \dots, (M_k)_{q_0^{(k)}}) \quad (k \in \mathbf{N}),$$

где

$$M_i = (Q_{M_i}, X_{M_i}, Y_{M_i}, \delta_{M_i}, \lambda_{M_i}) \quad (i = 1, \dots, k),$$

состоит из k допустимых инициальных автоматов, помещенных в начальный момент времени, соответственно, в вершины $v_0^{(1)}, \dots, v_0^{(k)}$ лабиринта $L = (V, E) \in \mathbf{L}(n)$ (такой лабиринт называется *инициальным* для системы M и обозначается через $L_{v_0^{(1)}, \dots, v_0^{(k)}}$). Предполагается, что входной алфавит автомата M_i ($i = 1, \dots, k$) – это множество

$$X_{M_i} = \Omega \times (\mathbf{B}(\Sigma) \setminus \{\emptyset\}) \times (Q_{M_1} \times \dots \times Q_{M_{i-1}} \times Q_{M_{i+1}} \times \dots \times Q_{M_k}).$$

Содержательно это означает, что *каждому автомату M_i ($i = 1, \dots, k$) в каждый момент времени доступна полная информация о состояниях всех остальных автоматов системы M в этот же момент времени*. Если при всех $i = 1, \dots, k$ каждая переменная, определяемая компонентой

$$Q_{M_1} \times \dots \times Q_{M_{i-1}} \times Q_{M_{i+1}} \times \dots \times Q_{M_k}$$

является *фиктивной* и для функции δ_{M_i} , и для функции λ_{M_i} , то система автоматов M называется *независимой*. В противном случае система автоматов M – *коллектив*. Таким образом, система M допустимых инициальных автоматов – *коллектив*, если существует такое $i = 1, \dots, k$, что для функции δ_{M_i} или для функции λ_{M_i} хотя бы одна из переменных, определяемых компонентой

$$Q_{M_1} \times \dots \times Q_{M_{i-1}} \times Q_{M_{i+1}} \times \dots \times Q_{M_k},$$

является *существенной*.

Поведением системы допустимых инициальных автоматов M в лабиринте $L_{v_0^{(1)}, \dots, v_0^{(k)}}$ называется упорядоченный набор последовательностей

$$(\pi((M_1)_{q_0^{(1)}}, L_{v_0^{(1)}}), \dots, \pi((M_k)_{q_0^{(k)}}, L_{v_0^{(k)}})).$$

Существует два варианта понятий *быть обходом* и *быть ловушкой* для системы $M = ((M_1)_{q_0^{(1)}}, \dots, (M_k)_{q_0^{(k)}})$ допустимых инициальных автоматов.

Для 1-го варианта система M *обходит* лабиринт $L_{v_0^{(1)}, \dots, v_0^{(k)}}$, если существует такое $i \in \{1, \dots, k\}$, что

$$Int((M_i)_{q_0^{(i)}}, L_{v_0^{(i)}}) = V$$

(в этом случае последовательность $pr_2 \pi((M_i)_{q_0^{(i)}}, L_{v_0^{(i)}})$ – *обход*) и лабиринт $L_{v_0^{(1)}, \dots, v_0^{(k)}}$ – *ловушка* для M , если

$$Int((M_i)_{q_0^{(i)}}, L_{v_0^{(i)}}) \subset V$$

для всех $i = 1, \dots, k$.

Для 2-го варианта система M *обходит* лабиринт $L_{v_0^{(1)}, \dots, v_0^{(k)}}$, если

$$\bigcup_{i=1}^k Int(M_{q_0^{(i)}}^{(i)}, L_{v_0^{(i)}}) = V$$

(в этом случае набор

$$(pr_2 \pi((M_1)_{q_0^{(1)}}, L_{v_0^{(1)}}), pr_2 \pi((M_2)_{q_0^{(2)}}, L_{v_0^{(2)}}), \dots, pr_2 \pi((M_k)_{q_0^{(k)}}, L_{v_0^{(k)}}))$$

последовательностей – *обход*) и лабиринт $L_{v_0^{(1)}, \dots, v_0^{(k)}}$ – *ловушка* для M , если

$$\bigcup_{i=1}^k Int(M_{q_0^{(i)}}^{(i)}, L_{v_0^{(i)}}) \subset V.$$

Отметим, что оба эти варианта совпадают, если $k = 1$.

Понятия *обходит* и *ловушка*, естественным образом, распространяются на системы не инициальных автоматов и/или не инициальные лабиринты. При этом, если система $M = ((M_1)_{q_0^{(1)}}, \dots, (M_k)_{q_0^{(k)}})$ обходит лабиринт $L_{v_0^{(1)}, \dots, v_0^{(k)}}$ при любом выборе начальных вершин $v_0^{(1)}, \dots, v_0^{(k)}$, то говорят, что M *сильно обходит лабиринт* $L_{v_0^{(1)}, \dots, v_0^{(k)}}$, а если лабиринт $L_{v_0^{(1)}, \dots, v_0^{(k)}}$ – ловушка для M при любом выборе начальных вершин $v_0^{(1)}, \dots, v_0^{(k)}$, то говорят, что $L_{v_0^{(1)}, \dots, v_0^{(k)}}$ – *сильная ловушка* для M .

Известны, по крайней мере, два различных способа усиления возможностей системы допустимых автоматов в лабиринте.

Первый способ состоит в использовании *камней*, предназначенных для временной отметки посещаемой автоматом вершины. Автомат может оставить камень в вершине (т.е. разметить вершину камнем) или передвинуть камень в другую вершину. Камень может быть собственностью конкретного автомата (т.е. остальные автоматы системы не имеют права передвигать его) или быть коллективной собственностью некоторого подмножества автоматов системы (т.е. любой автомат из этого подмножества может передвигать камень). В последнем случае для разрешения конфликта устанавливается приоритет на право использовать камень. Если система состоит из k допустимых автоматов и m камней, говорят, что система допустимых автоматов имеет $\min(k, m)$.

Второй способ состоит в использовании более общих объектов, чем обычные конечные автоматы, т.е. недетерминированных автоматов, автоматов со счетчиками, автоматов с магазинной памятью и т.д., вплоть до машин Тьюринга, удовлетворяющих тем или иным условиям.

В процессе исследования поведения автоматов в лабиринтах основное внимание уделяется проблемам *анализа* и *синтеза*.

Проблемы анализа сводятся к следующим двум формулировкам:

1) по заданной системе автоматов описать классы всех тех (возможно, удовлетворяющих тем или иным ограничениям) лабиринтов, которые эта система либо обходит, либо которые являются для нее ловушкой;

2) по заданному лабиринту, описать классы всех тех (возможно, удовлетворяющих тем или иным ограничениям) систем автоматов, которые либо обходят этот лабиринт, либо для которых он является ловушкой.

Проблемы синтеза сводятся к следующим двум формулировкам:

1) описать классы всех тех (возможно, удовлетворяющих тем или иным ограничениям) систем автоматов, которые либо обходят любой из лабиринтов, принадлежащих заданному классу, либо для которых любой из лабиринтов, принадлежащих заданному классу, является ловушкой;

2) описать классы всех тех (возможно, удовлетворяющих тем или иным ограничениям) лабиринтов, которые либо обходит любая система автоматов, принадлежащая заданному классу, либо которые являются ловушкой для любой системы автоматов, принадлежащей заданному классу.

Перечислим кратко некоторые результаты, связанные с исследованием поведения автоматов в лабиринтах, приведенные в обзоре [6] (содержащем достаточно полный анализ результатов, полученных до 1992).

В 1978 году *L. Budach* установил следующий факт.

Теорема 1.15. Не существует конечного инициального автомата, который сильно обходит все мозаичные плоские конечные лабиринты.

Для плоских шахматных конечных не инициальных лабиринтов установлена несколько иная ситуация (см., напр., [6]).

Теорема 1.16. Не существует алгоритма, устанавливающего по заданному автомату, сильно ли он обходит любой плоский шахматный конечный односвязный лабиринт.

Поэтому естественным является построение автоматов, обходящих все лабиринты, принадлежащие подклассам плоских шахматных конечных лабиринтов, удовлетворяющих специальным ограничениям. В 1992 году С.А. Богомолов, А.А. Золотых и А.Н. Зыричев установили следующий (по-видимому, наиболее сильный) результат в этом направлении.

Теорема 1.17. Для любых $d \in \mathbf{R}_+$, $l \in \mathbf{Z}$, $m \in \mathbf{N}$ существует инициальный автомат, имеющий не более чем $c \cdot d$ состояний ($c > 0$ – константа), который обходит класс всех таких плоских шахматных конечных инициальных лабиринтов, что каждое множество

$$\bigcup_{(i,j) \in D} Kv(i,j)$$

лежит в полосе ширины d с тангенсом угла наклона, равным $l \cdot m^{-1}$ (где D – конечная дыра, а $Kv(i,j)$ – единичный квадрат в \mathbf{R}^2 с центром в точке $(i,j) \in \mathbf{Z}^2$).

В 1990 году Г. Килибарда обобщил теорему 1.15 следующим образом.

Теорема 1.18. Не существует независимой конечной системы инициальных конечных автоматов $M = ((M_1)_{q_0^{(1)}}, \dots, (M_k)_{q_0^{(k)}})$, обходящей все мозаичные плоские конечные инициальные лабиринты $L_{\underbrace{v_0, \dots, v_h}_{k \text{ раз}}}$.

Если v_0, v_1, \dots – обход автоматом конечного лабиринта, то всегда существует такое $h \in \mathbf{N}$, что начальный отрезок v_0, v_1, \dots, v_h – обход. Поэтому естественно возникает *проблема остановки* автомата по завершению обхода конечного лабиринта. В 1990 году М. Bull и А. Hemmerling выделили следующие случаи, когда эта проблема имеет отрицательное решение.

Теорема 1.19. Не существует инициальный автомат, который обходит и останавливается по завершению обхода каждого лабиринта из класса всех плоских конечных:

- 1) шахматных односвязных лабиринтов;
- 2) мозаичных лабиринтов, являющихся деревьями.

Усилить возможности для обхода лабиринтов можно за счет использования *камней*. В 1984 году К. Kriegel установил следующий факт.

Теорема 1.20. Для любого $k \in \mathbb{N}$ существует коллектив типа (1,1), который сильно обходит все мозаичные плоские конечные лабиринты с не более, чем k дырами.

В 1986 году *A. Hemmerling* доказал, что существует указанный в теореме 1.20 коллектив типа (1,1), где автомат имеет не более, чем c^k ($c > 0$ – константа) состояний.

Естественно возникает задача описания множества минимальных типов коллективов, обходящих все лабиринты из заданного класса. В 1992 году *Г. Килибарда* получил следующее частичное решение этой задачи.

Теорема 1.21. Типы (2,3), (3,2), (4,1) и (5,0) являются минимальными типами коллективов, для которых существует обход классов как всех плоских мозаичных лабиринтов, так и всех плоских мозаичных лабиринтов, не содержащих бесконечных дыр.

Теоремы 1.18-1.21 характеризуют усиление возможностей обхода плоских лабиринтов за счет перехода от независимых систем автоматов к коллективам автоматов. Результаты, которые получили *M. Blum* и *W. Sakoda* в 1977 году и *A. Hemmerling* в 1987 году следующим образом характеризуют возможности обхода коллективам автоматов лабиринтов из $\mathbf{L}(n)$ в случае, когда $n \geq 3$.

Теорема 1.22. Не существует коллектива автоматов, обходящего все 3-мерные мозаичные конечные инициальные лабиринты.

Из теорем 1.18 и 1.22 вытекает, что представляет интерес построение мозаичных универсальных ловушек для допустимых систем автоматов. Обзор таких ловушек дан в [6].

Усилить возможности обхода лабиринтов можно за счет использования более общих объектов, чем обычные конечные автоматы. В 1978 году *W. Соу* установил следующие характеристики возможностей автоматов с магазином и линейно ограниченных МТ при обходе плоских лабиринтов.

Теорема 1.23. 1. Не существует автомат с магазином, который обходит все плоские конечные 3-регулярные графы. 2. Существует линейно ограниченная МТ, которая обходит все плоские конечные 3-регулярные графы. 3. Существует автомат с магазином, который обходит все мозаичные плоские конечные лабиринты.

В 1990 году *M. Bull* и *A. Hemmerling* выделили подкласс мозаичных плоских лабиринтов, для обхода элементов которого возможности автомата с магазином следующим образом усиливаются по сравнению с возможностями, установленными в теореме 1.23 (п.3).

Теорема 1.24. Существует автомат с магазином, который обходит все плоские шахматные односвязные конечные лабиринты и останавливается каждый раз по завершению обхода лабиринта.

В 1987 году *A. Hemmerling* показал, что возможности многоголовочных автоматов при обходе лабиринтов выше, чем возможности коллективов автоматов. Ряд результатов, связанных с описанием множеств минимальных типов коллективов, которые содержат автоматы со счетчиками или автоматы с магазинами, установил в том же, 1987 году, *A.B. Анджанс*.

Выше говорилось об обходе лабиринта системой автоматов. Однако этим не исчерпывается проблематика исследования поведения автоматов в лабиринтах. Следует отметить различные варианты задачи о встрече систем автоматов в лабиринте, исследование вычислимости системами автоматов предикатов, определенных на множестве лабиринтов того или иного типа и проблему различимости лабиринтов, принадлежащих заданному классу, системами автоматов, удовлетворяющих заданным ограничениям.

В заключение отметим следующий момент, характеризующий высокую внутреннюю сложность проблем исследования поведения автоматов в лабиринтах. Любую систему конечных автоматов в лабиринте можно рассматривать как (возможно) удовлетворяющую тем или иным ограничениям (возможно, многоголовочную) МТ с k управляющими устройствами и одной не ограниченной n -мерной лентой (и, возможно, дополнительными рабочими лентами). Начальная конфигурация для такой МТ является *растровым* представлением исследуемого лабиринта, принадлежащего множеству $\mathbf{L}(n)$: вершины представлены клетками, а наличие или отсутствие ребра, параллельного заданной координатной оси, легко реализуется за счет допущения о многоэтажности ленты (очевидно, что для представления любого лабиринта $L \in \mathbf{L}(n)$ достаточен $n+1$ этаж на ленте). Существенное отличие от классической теории алгоритмов состоит в том, что активная зона начальной конфигурации может состоять из бесконечного числа клеток. Как следствие, в этом случае не может быть непосредственно применен ни один результат классической теории алгоритмов, существенно опирающийся на предположение о конечности активной зоны конфигурации. Кроме того, приведенная интерпретация показывает, что существование обхода бесконечного лабиринта системой автоматов эквивалентно существованию такого бесконечного процесса

$$\mathbf{K} : K = K_1, K_2, \dots$$

переработки МТ начальной конфигурации K с бесконечной активной зоной, что каждая непустая клетка конфигурации K хотя бы раз обзревается хотя бы одной головкой. Эта проблема вообще не является предметом исследования классической теории алгоритмов.

1.4. Основные понятия и определения.

Представление графа перечислением его вершин и ребер дает возможность в полной мере использовать язык теории множеств для изложения результатов теории графов. Однако этот способ не приемлем в процессе исследования алгоритмов на графах. Действительно, анализ любого алгоритма осуществляется в терминах той или иной *модели вычислений* (выбранной хотя бы на концептуальном уровне), что автоматически означает, представление множеств вершин и ребер графа в терминах этой модели. Для любой модели вычислений простое перечисление множества всех ребер графа делает весьма сложным выбор *окрестности вершины* графа – объекта, играющего существенную роль во всех алгоритмах на графах, в которых используется операции вида «выбрать ребро, инцидентное данной вершине», «выбрать вершину, смежную данной вершине», «выбрать ребро, смежное данному ребру».

Представление графа прямоугольным k -мерным ($k \in \mathbf{N}$) лабиринтом имеет некоторую аналогию с графическим способом задания графа и дает возможность непосредственно использовать в качестве модели вычислений k_1 -мерную ($k_1 \in \mathbf{N}$) (возможно, многоголовочную) МТ, где $k_1 \geq k$. Однако, как показывает следующая лемма, кроме сложностей, непосредственно связанных с исследованием поведения автоматов в лабиринтах, не всякий граф может быть представлен прямоугольным k -мерным лабиринтом ($k \in \mathbf{N}$).

Лемма 1.1. При фиксированном множестве вершин V число $f_{|V|}$ графов $G = (V, E)$ ($E \subseteq V^{(2)}$), которые могут быть представлены прямоугольными k -мерными лабиринтами $L \in \bigcup_{k=1}^{|V|-1} \mathbf{L}(k)$, удовлетворяет неравенству

$$|V|^{|V|-2} \leq f_{|V|} \leq 2^{\binom{|V|}{2}} - \frac{1}{|V|} \cdot \sum_{i=1}^{|V|-1} i \cdot 2^{\binom{|V|-i}{2}} f_i.$$

Доказательство. По определению, любой лабиринт $L \in \mathbf{L}(k)$ ($k \in \mathbf{N}$) является связным графом. Поэтому справедливость неравенства

$$f_{|V|} \leq 2^{\binom{|V|}{2}} - \frac{1}{|V|} \cdot \sum_{i=1}^{|V|-1} i \cdot 2^{\binom{|V|-i}{2}} f_i$$

вытекает из оценки числа связных графов (см., напр. [12]). С другой стороны при подходящем значении $k \in \{1, \dots, |V| - 1\}$ любое дерево с множест-

вом вершин V может быть представлено прямоугольным k -мерным лабиринтом. Справедливость неравенства

$$f_{|V|} \geq |V|^{|V|-2}$$

вытекает из оценки числа деревьев с множеством вершин V .

Лемма доказана.

Следующая теорема описывает нетривиальный класс графов, которые не могут быть представлены прямоугольными k -мерными лабиринтами ни при каком $k \in \mathbf{N}$.

Теорема 1.25. Ни один граф $G = (V, E)$, содержащий подграф, являющийся треугольником, не может быть представлен прямоугольным k -мерным лабиринтом ни при каком $k \in \mathbf{N}$.

Доказательство. Предположим противное, т.е. что граф $G = (V, E)$, содержащий подграф, являющийся треугольником, определяемый вершинами v_1, v_2 и v_3 , может быть представлен прямоугольным лабиринтом $L \in \mathbf{L}(k)$ при некотором $k \in \mathbf{N}$. Рассмотрим точки $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3 \in \mathbf{R}^k$, представляющие, соответственно вершины v_1, v_2 и v_3 графа G . Из определения прямоугольного k -мерного лабиринта вытекает, что существует единственная i -я координата ($1 \leq i \leq k$), по которой различаются точки \mathbf{v}_1 и \mathbf{v}_2 , и существует единственная j -я координата ($1 \leq j \leq k, j \neq i$), по которой различаются точки \mathbf{v}_1 и \mathbf{v}_3 . Но тогда точки \mathbf{v}_2 и \mathbf{v}_3 различаются и по i -й, и по j -й координатам. Это противоречит тому, что ребро $\{v_2, v_3\}$ представлено в \mathbf{R}^k отрезком, параллельным координатной оси. Полученное противоречие показывает, что предположение – ложное, т.е. граф, содержащий подграф, являющийся треугольником, не может быть представлен прямоугольным k -мерным лабиринтом ни при каком $k \in \mathbf{N}$.

Теорема доказана.

Следствие 1.1. Размер и порядок любого графа $G = (V, E)$, который может быть представлен прямоугольным k -мерным лабиринтом ($k \in \mathbf{N}$), удовлетворяет неравенству

$$e(G) \leq \lfloor 0.25 \cdot |G|^2 \rfloor.$$

Доказательство. В силу теоремы 1.25, если граф $G = (V, E)$ может быть представлен прямоугольным k -мерным лабиринтом ($k \in \mathbf{N}$), то граф G не содержит подграф, являющийся треугольником. Из теоремы 1.8 вы-

текает, что если граф $G = (V, E)$ не содержит подграф, являющийся треугольником, то $e(G) \leq \lfloor 0.25 \cdot |G|^2 \rfloor$.

Следствие доказано.

Пример 1.4. Граф G_1 из примера 1.1 содержит треугольники (один из них определяется вершинами v_2, v_4 и v_5 , а другой – вершинами v_3, v_4 и v_5). Следовательно, граф G_1 не может быть представлен прямоугольным k -мерным лабиринтом ни при каком $k \in \mathbf{N}$.

Таким образом, естественно выбрать в качестве представления графа либо соответствующую матрицу (смежности вершин, смежности ребер или инциденций), либо списки смежности (вершин или ребер), что обеспечивает как универсальность представления, так и возможность для эффективной работы с окрестностями вершин. В качестве модели вычислений естественно выбрать многомерную (возможно, многоголовочную и многоленочную) МТ.

Обозначим через $\mathbf{G}(n, m)$ ($n \in \mathbf{N}, m \in \{0, 1, \dots, \binom{n}{2}\}$) – множество всех (n, m) -графов $G = (V, E)$ с множеством вершин $V = \{1, \dots, n\}$. Зафиксируем представление $\mathbf{R}(G)$ графа $G \in \mathbf{G}(n, m)$ либо матрицей (смежности вершин, смежности ребер или инциденций), либо списками смежности (вершин или ребер). В тех случаях, когда граф $G = (V, E)$ представлен матрицей смежности ребер, матрицей инциденций или списками смежности ребер, считаем, что $E = \{1, \dots, m\}$. Всюду в дальнейшем представление $\mathbf{R}(G)$ рассматривается в качестве исходных данных k -мерной (возможно, многоголовочной и многоленочной) МТ $M = (Q, A, I, \delta, \Lambda, q_{in}, q_{fin})$. При этом предполагается, что рабочий алфавит A МТ M содержит специальный маркер $\#$, применяемый, в частности, для выделения зоны, используемой для представления исходных данных МТ. Если в процессе исследования сложности алгоритмов для графов $G \in \mathbf{G}(n, m)$ выбран *равномерный вес*, то предполагается, что:

1) если $\mathbf{R}(G)$ ($G \in \mathbf{G}(n, m)$) – представление графа G матрицей смежности вершин или списками смежности вершин, то $V \subset A$ и элементы множества V непосредственно заданы в виде чисел $1, \dots, n$;

2) если $\mathbf{R}(G)$ ($G \in \mathbf{G}(n, m)$) – представление графа G матрицей смежности ребер или списками смежности ребер, то $E \subset A$ и элементы множества E непосредственно заданы в виде чисел $1, \dots, m$;

3) если $\mathbf{R}(G)$ ($G \in \mathbf{G}(n, m)$) – представление графа G матрицей инциденций, то $\{1, \dots, \max\{n, m\}\} \subset A$ и элементы множества V непосредст-

венно заданы в виде чисел $1, \dots, n$, а элементы множества E – в виде чисел $1, \dots, m$.

Аналогичным образом, если в процессе исследования сложности алгоритмов для графов $G \in \mathbf{G}(n, m)$ выбран *логарифмический вес*, то предполагается, что $\{0, 1\} \subset A$, причем:

1) если $\mathbf{R}(G)$ ($G \in \mathbf{G}(n, m)$) – представление графа G матрицей смежности вершин или списками смежности вершин, то элементы множества V заданы в виде двоичных последовательностей длины $\lceil \log n \rceil$;

2) если $\mathbf{R}(G)$ ($G \in \mathbf{G}(n, m)$) – представление графа G матрицей смежности ребер или списками смежности ребер, то и элементы множества E заданы в виде двоичных последовательностей длины $\lceil \log m \rceil$;

3) если $\mathbf{R}(G)$ ($G \in \mathbf{G}(n, m)$) – представление графа G матрицей инцидентностей, то элементы множеств V и E заданы в виде двоичных последовательностей длин, соответственно, $\lceil \log n \rceil$ и $\lceil \log m \rceil$.

Пример 1.5. Рассмотрим граф G_1 из примера 1.1. В качестве модели вычислений выбрана одноголовочная двумерная МТ с одной лентой типа \mathbf{Z}^2 . На рис. 1.8 показаны представления графа G_1 в случае, когда выбран равномерный вес, а на рис. 1.9 и 1.10 в случае, когда выбран логарифмический вес, представления графа G_1 , соответственно, матрицами и списками смежности.

Замечание 1.1. 1. Предложенный на рис. 1.10 способ представления матриц (n, m) -графа основан на непосредственном блочном моделировании МТ с произвольным внешним алфавитом МТ с внешним алфавитом $A = \{0, 1, \Lambda, \#\}$.

2. Объем памяти, необходимой для представления (n, m) -графа в виде исходных данных двумерной МТ при логарифмическом весе, способом, предложенным на рис. 1.10, для матрицы смежности вершин, матрицы смежности ребер и матрицы инцидентностей равен, соответственно, $O(n^2 \log n)$ ($n \rightarrow \infty$), $O(m^2 \log m)$ ($n, m \rightarrow \infty$) и $O(n(m \log m + \log n))$ ($n, m \rightarrow \infty$). Этот объем можно снизить, соответственно, до величины $O(n^2)$ ($n \rightarrow \infty$), $O(m^2)$ ($n, m \rightarrow \infty$) и $O(mn)$ ($n, m \rightarrow \infty$), если отказаться от представления имен вершин и/или ребер графа. Фиксация начала зоны представления графа и предположение о том, что $V = \{1, \dots, n\}$ и $E = \{1, \dots, m\}$ дает возможность легко вычислять и имена вершин, и имена ребер.

Замечание 1.2. При блочном моделировании МТ переход от равномерного веса к логарифмическому происходит с логарифмическим замедлением во времени (см. теорему 1.1) и с логарифмическим увеличением объема памяти (см. замечание 1.1). Это означает, что если при равномерном весе временная и емкостная сложность алгоритма A равны, соответственно, $T_A(n, m)$ и $V_A(n, m)$, то при переходе к логарифмическому весу временная и емкостная сложность алгоритма A равны, соответственно, $T_A(n, m) \cdot \max\{\log m, \log n\}$ и $V_A(n, m) \cdot \max\{\log m, \log n\}$. Учитывая сказанное, в

дальнейшем будем считать, что при исследовании алгоритмов на *графах всегда выбран равномерный вес* (если в явном виде не оговорено противное), что упрощает изложение и не ограничивает общность рассуждений.

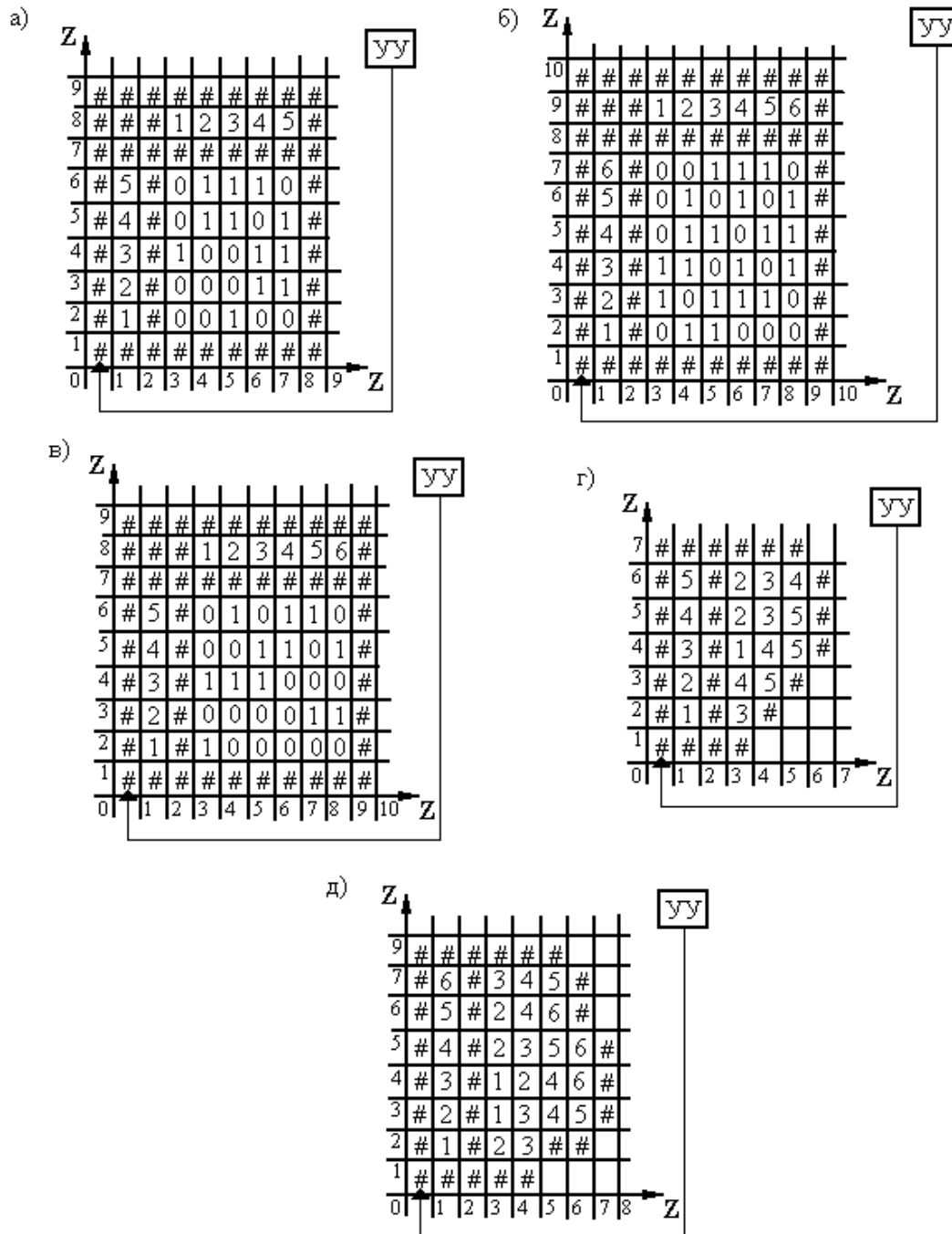


Рис. 1.8. Представление графа G_1 из примера 1.1 исходными данными двумерной МТ при равномерном весе: а) матрицей смежности вершин; б) матрицей смежности ребер; в) матрицей инцидентий; г) списками смежности вершин; д) списками смежности ребер.

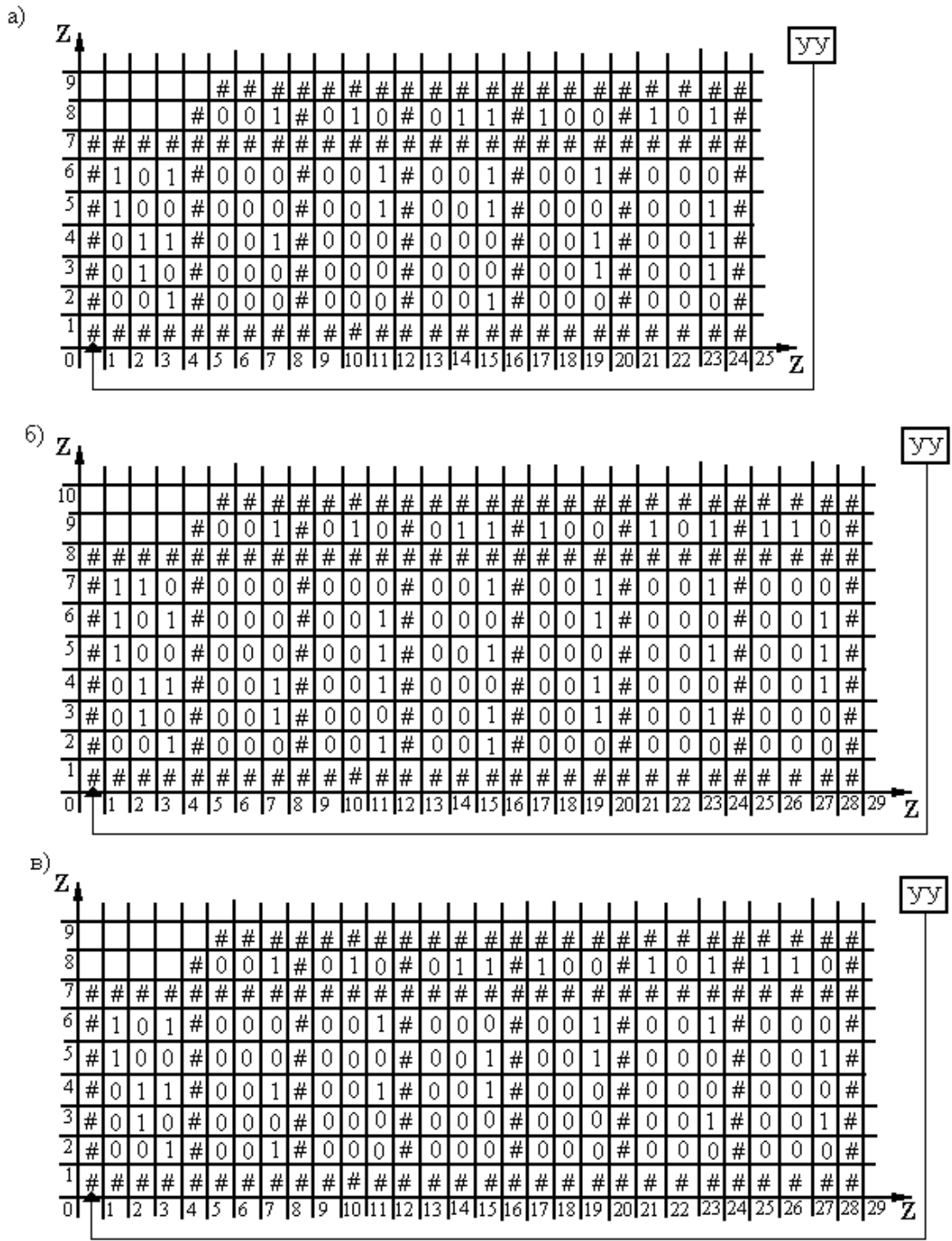


Рис. 1.9. Представление графа G_1 из примера 1.1 исходными данными двумерной МГ при логарифмическом весе: а) матрицей смежности вершин; б) матрицей смежности ребер; в) матрицей инцидентий.

4) если $\mathbf{R}(G)$ – представление графа $G \in \mathbf{G}(n, m)$ списками смежности вершин, то $V(\mathbf{R}, n, m) = O(\max\{n, m\})$ ($n, m \rightarrow \infty$);

5) если $\mathbf{R}(G)$ – представление графа $G \in \mathbf{G}(n, m)$ списками смежности ребер, то $V(\mathbf{R}, n, m) = O(m \cdot \max\{1, \min\{n, m\}\})$ ($n, m \rightarrow \infty$).

Обозначим через $v_A(\mathbf{R}, G)$ ($G \in \mathbf{G}(n, m)$) сигнализирующую емкости алгоритма A (реализованного МТ выбранного типа) при работе с представлением $\mathbf{R}(G)$ графа G и положим $v_A(\mathbf{R}, n, m) = \max_{G \in \mathbf{G}(n, m)} v_A(\mathbf{R}, G)$. Следуя традиционному подходу в теории алгоритмов, в дальнейшем будем работать с емкостной сложностью $V_A(\mathbf{R}, n, m)$ алгоритма A , т.е. с асимптотической оценкой функции $v_A(\mathbf{R}, n, m)$.

Определение 1.1. Назовем что алгоритм A *локальным* для множества графов $\mathbf{G}(n, m)$ и представления \mathbf{R} , если

$$V_A(\mathbf{R}, n, m) = O(V(\mathbf{R}, n, m)) \quad (n, m \rightarrow \infty).$$

Очевидно, что для алгоритма A свойство «быть локальным» для множества графов $\mathbf{G}(n, m)$ и представления \mathbf{R} существенно зависит от того, по какому закону осуществляется рост $m \rightarrow \infty$. Ясно, также, что если представляет интерес исследование свойства «быть локальным» для алгоритма A при любом (допустимом) законе роста $m \rightarrow \infty$, то естественно перейти от множества $\mathbf{G}(n, m)$ к множеству $\bigcup_{m=0}^n \mathbf{G}(n, m)$.

Пример 1.7. Выберем в качестве исходных данных одноголовочной двумерной МТ с одной лентой типа \mathbf{Z}^2 представление $\mathbf{R}(G)$ графа $G \in \mathbf{G}(n, m)$ матрицей смежности вершин

$$\mathbf{M}_G = \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{pmatrix}.$$

Тогда (см. пример 1.6)

$$V(\mathbf{R}, n, m) = O(n^2) \quad (n \rightarrow \infty).$$

Покажем, что проверка свойства «*быть связным графом*» осуществима с помощью алгоритма, локального для представления \mathbf{R} при любом (допустимом) законе роста $m \rightarrow \infty$.

Обозначим через $f(\mathbf{M}_G)$ матрицу, полученную в результате перестановки как строк, так и столбцов матрицы \mathbf{M}_G в соответствии с подстановкой $f \in S(n)$, т.е. i -я строка (соответственно, i -й столбец) матрицы $f(\mathbf{M}_G)$ совпадает с $f(i)$ -й строкой

(соответственно, с $f(i)$ -м столбцом) матрицы \mathbf{M}_G при всех $i = 1, \dots, n$. Ясно, что граф G связан тогда и только тогда, когда ни одна из матриц $f(\mathbf{M}_G)$ ($f \in S(n)$) не имеет клеточную форму вида

$$\mathbf{A} = \begin{pmatrix} \mathbf{A}_1 & \mathbf{O}_{12} & \dots & \mathbf{O}_{1k} \\ \mathbf{O}_{21} & \mathbf{A}_2 & \dots & \mathbf{O}_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{O}_{k1} & \mathbf{O}_{k2} & \dots & \mathbf{A}_k \end{pmatrix} \quad (k \geq 2), \quad (*)$$

где \mathbf{O}_{ij} ($i, j \in \{1, \dots, k\}, i \neq j$) – нулевая матрица соответствующего порядка.

Рассмотрим алгоритм A проверки свойства «быть связным графом», каждая итерация которого состоит в следующем ($FLAG$ – флаг (т.е. булева переменная), определяющий истинность высказывания « G – связный граф»):

Шаг 1. Порождается очередной элемент $f \in S(n)$.

Шаг 2. Строится матрица $f(\mathbf{M}_G)$.

Шаг 3. Если матрица $f(\mathbf{M}_G)$ имеет вид (*), то $FLAG := 0$, иначе $FLAG := 1$.

Последовательное порождение (одного за другим) элементов множества $S(n)$ осуществимо в рамках памяти размера $O(n^2)$ ($n \rightarrow \infty$), что иллюстрируется, следующей процедурой

Procedure PRMTTNS_1(n).

Шаг 1. $\alpha := \underbrace{0 \dots 0}_{n^2 \text{ раз}}$.

Шаг 2. Разбивая слева направо последовательность α на отрезки длины n , расположить их в виде строк матрицы \mathbf{B} .

Шаг 3. Если матрица \mathbf{B} содержит в каждой строке и в каждом столбце в точности по одной единице, то порожден очередной элемент множества $S(n)$ и *конец*, иначе переход к шагу 4.

Шаг 4. $\alpha := \alpha + 1$.

Шаг 5. Если $\alpha = \underbrace{1 \dots 1}_{n^2 \text{ раз}}$, то все элементы множества $S(n)$ уже были порождены и *конец*, иначе переход к шагу 2.

Пусть в результате работы процедуры PRMTTNS_1(n) порождена матрица $\mathbf{B} = (b_{ij})$, определяющая подстановку $f \in S(n)$. Тогда матрица $f(\mathbf{M}_G)$ может быть вычислена по формуле

$$f(\mathbf{M}_G) = \mathbf{B} \cdot \mathbf{M}_G \cdot \mathbf{B}^T,$$

где \mathbf{B}^T – матрица, транспонированная к матрице \mathbf{B} . Таким образом, построение матрицы $f(\mathbf{M}_G)$ осуществимо в рамках памяти размера $O(n^2)$ ($n \rightarrow \infty$).

Проверка матрицы $f(\mathbf{M}_G) = (c_{ij})$ на вид (*) также осуществима в рамках памяти размера $O(n^2)$ ($n \rightarrow \infty$), что иллюстрируется следующими действиями, реализующими схему, основанную на идее, представленной на рис. 1.11:

Шаг 1. $cntr := 1, i := 1, l := 1$.

Шаг 2. Если $i < n$, то переход к шагу 3, иначе – к шагу 6.

Шаг 3. Если $c_{i+l+j, i+h} = 0$ и $c_{i+h, i+l+j} = 0$ для всех $j = 0, 1, \dots, n-i-l$ и $h = 0, 1, \dots, l-1$, то $cntr := cntr + 1, i := i + l, l := 1$ переход к шагу 2, иначе $l := l + 1$ – к шагу 5.

Шаг 5. Если $l \leq n - i$, то переход к шагу 3, иначе – переход к шагу 6.

Шаг 6. Если $cntr \geq 2$, то матрица $f(\mathbf{M}_G) = (c_{ij})$ имеет вид (*) и *конец*, иначе матрица $f(\mathbf{M}_G) = (c_{ij})$ не имеет вид (*) и *конец*.

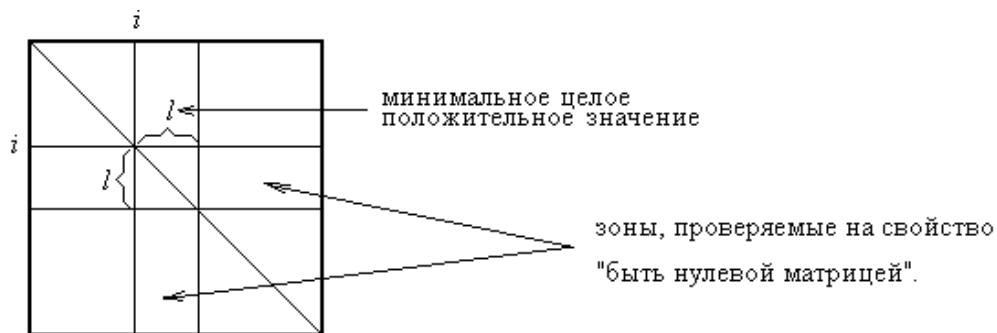


Рис. 1.11. Идея схемы проверки матрицы $f(\mathbf{M}_G)$ на вид (*).

Таким образом, емкостная сложность алгоритма A равна

$$V_A(\mathbf{R}, n, m) = O(n^2) + O(n^2) + O(n^2) + O(n^2) = O(n^2) \quad (n, m \rightarrow \infty).$$

А так как $v_A(\mathbf{R}, n, m) \leq c \cdot v(\mathbf{R}, n, m)$, где c – положительная константа, то

$$V_A(\mathbf{R}, n, m) = O(V(\mathbf{R}, n, m)) \quad (n, m \rightarrow \infty),$$

т.е. проверка свойства «*быть связным графом*» осуществима с помощью алгоритма, локального для представления \mathbf{R} , что и требовалось показать.

Пример 1.8. Как и в предыдущем примере, выберем в качестве исходных данных одноголовочной двумерной МТ с одной лентой типа \mathbf{Z}^2 представление $\mathbf{R}(G)$ графа $G \in \mathbf{G}(n, m)$ матрицей смежности вершин.

Проверка того, являются ли заданные графы $G_1, G_2 \in \mathbf{G}(n, m)$, осуществима с помощью алгоритма, локального для представления \mathbf{R} при любом (допустимом) законе роста $m \rightarrow \infty$.

Действительно, достаточно последовательно (одну за другой) порождать, подстановки $f \in S(n)$ (например, с помощью процедуры PRMTTNS_1(n), приведенной в примере 1.7) и проверять: истинно ли равенство

$$f(\mathbf{M}_{G_1}) = \mathbf{M}_{G_2} ?$$

Если существует такая подстановка $f \in S(n)$, что равенство $f(\mathbf{M}_{G_1}) = \mathbf{M}_{G_2}$ – истинное, то графы G_1 и G_2 – изоморфные. В противном случае графы G_1 и G_2 изоморфными не являются.

Для предложенного алгоритма A истинно неравенство

$$v_A(\mathbf{R}, n, m) \leq c \cdot v(\mathbf{R}, n, m),$$

где c – положительная константа. Следовательно,

$$V_A(\mathbf{R}, n, m) = O(V(\mathbf{R}, n, m)) \quad (n, m \rightarrow \infty),$$

т.е. проверка того, являются ли изоморфными заданные графы $G_1, G_2 \in \mathbf{G}(n, m)$, осуществима с помощью алгоритма, локального для представления \mathbf{R} , что и требовалось показать.

Пример 1.9. Выберем в качестве исходных данных одноголовочной двумерной МТ с одной лентой типа \mathbf{Z}^2 представление $\mathbf{R}(G)$ графа $G \in \mathbf{G}(n, m)$ списками смежности вершин. Тогда (см. пример 1.6)

$$V(\mathbf{R}, n, m) = O(\max\{n, m\}) \quad (n, m \rightarrow \infty).$$

Покажем, что и в этом случае проверка того, являются ли изоморфными заданные графы $G_1, G_2 \in \mathbf{G}(n, m)$, также (как и в примере 1.8) осуществима с помощью алгоритма, локального для представления \mathbf{R} при любом (допустимом) законе роста $m \rightarrow \infty$.

Обозначим через $f(\mathbf{R}(G))$ ($f \in S(n), G \in \mathbf{G}(n, m)$) списки смежности вершин, полученные из списков смежности вершин $\mathbf{R}(G)$ в результате следующих действий: каждый элемент $i \in \{1, \dots, n\}$ каждого списка заменяется элементом $f(i)$, а затем порядок списков изменяется в соответствии с подстановкой $f \in S(n)$, т.е. j -й список становится $f(j)$ -м списком. Ясно, что построение списков смежности вершин $f(\mathbf{R}(G))$ осуществимо в рамках памяти размера $O(n)$ ($n \rightarrow \infty$).

Проверка равенства

$$f(\mathbf{R}(G_1)) = \mathbf{R}(G_2)$$

сводится к проверке совпадения двух систем множеств и, следовательно, осуществима в рамках памяти размера $O(\max\{n, m\})$ ($n, m \rightarrow \infty$).

Следовательно, достаточно показать, что последовательное порождение (одного за другим) элементов множества $S(n)$ осуществимо в рамках памяти размера $O(n)$ ($n \rightarrow \infty$). Сопоставив с подстановкой $f \in S(n)$ последовательность

$$f(1)f(2)\dots f(n),$$

установим взаимно-однозначное соответствие между множеством $S(n)$ и множеством последовательностей

$$A = \{a_1 \dots a_n \mid a_i \in \{1, \dots, n\} (i = 1, \dots, n) \text{ и } a_i \neq a_j \text{ при } i \neq j\}.$$

Определим на множестве A обычное отношение лексикографического порядка \prec , т.е.

$$\begin{aligned} a_1 \dots a_n \prec b_1 \dots b_n &\Leftrightarrow \\ \Leftrightarrow (\exists i \in \{0, 1, n-1\})(\forall j = 1, \dots, i)(a_j = b_j \ \& \ a_{i+1} < b_{i+1}). \end{aligned}$$

Элемент множества A , непосредственно следующий (относительно лексикографического порядка \prec) за элементом $a_1 \dots a_n \in A$, может быть порожден в рамках памяти размера $O(n)$ ($n \rightarrow \infty$), что иллюстрируется следующими действиями, реализующими схему, основанную на идее, представленной на рис. 1.12 (через γ^{-1} обозначена последовательность γ , записанная в обратном порядке):

Procedure PRMTTNS_2($a_1 \dots a_n$).

Шаг 1. Если $a_1 \dots a_n = n(n-1) \dots 21$, то следующей подстановки не существует и *конец*, иначе $i := n-1$ и переход к шагу 2.

Шаг 2. Если $a_i > a_{i+1}$, то $i := i-1$ и переход к шагу 2, иначе – к шагу 3.

Шаг 3. $\zeta := a_1 \dots a_{i-1}$, $j := i+1$.

Шаг 4. Если $a_j > a_i$, то переход к шагу 5, иначе $j := j-1$ и переход к шагу 9.

Шаг 5. $j := j+1$.

Шаг 6. Если $j \leq n$, то переход к шагу 4, иначе – к шагу 7.

Шаг 7. $\beta := a_{i+1} \dots a_n$.

Шаг 8. Выдать последовательность $\zeta\beta^{-1}a_i$ в качестве порожденной подстановки и *конец*.

Шаг 9. $\xi := a_{i+1} \dots a_{j-1}$, $\beta := a_{j+1} \dots a_n$.

Шаг 10. Выдать последовательность $\zeta a_j \beta^{-1} a_i \xi^{-1}$ в качестве порожденной подстановки и *конец*.

Рассмотрим алгоритм A проверки того, являются ли изоморфными заданные графы $G_1, G_2 \in \mathbf{G}(n, m)$, каждая итерация которого состоит в следующем ($FLAG$ – флаг (т.е. булева переменная) определяющий истинность высказывания «графы G_1 и G_2 – изоморфные»):

Шаг 1. Порождается очередной элемент $f \in S(n)$

Шаг 2. Строится представление $f(\mathbf{R}(G_1))$.

Шаг 3. Если $f(\mathbf{R}(G_1)) = \mathbf{R}(G_2)$, то $FLAG := 1$, иначе $FLAG := 0$.

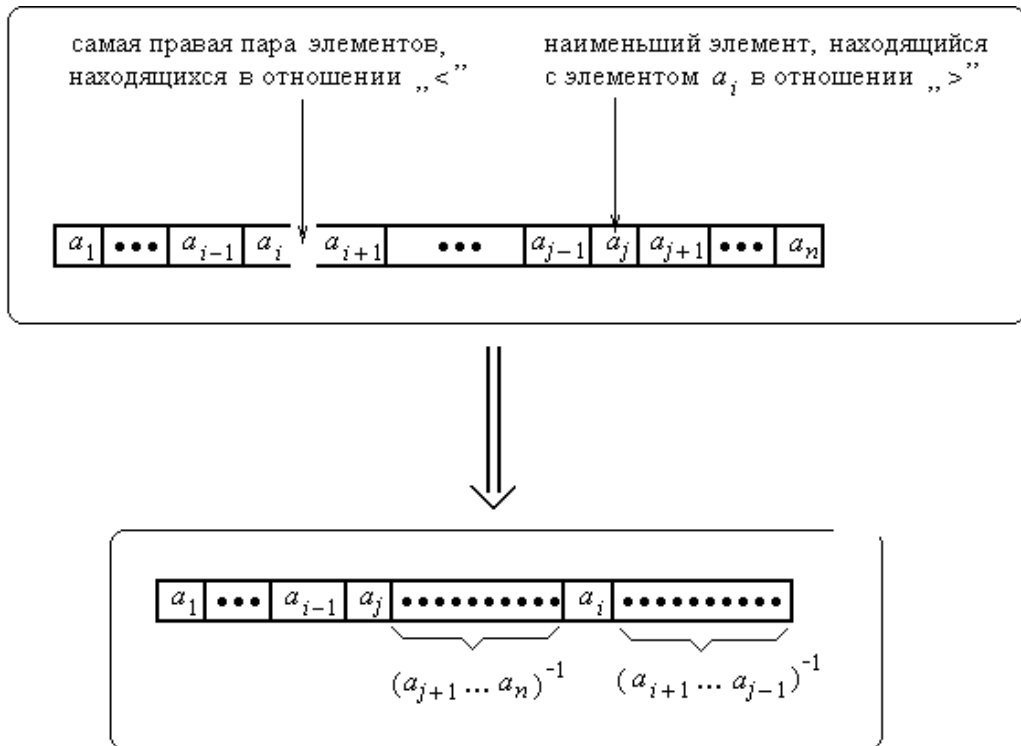


Рис. 1.12. Идея схемы порождения следующего элемента множества A .

Емкостная сложность предложенного алгоритма A равна

$$V(\mathbf{R}, n, m) = O(\max\{n, m\}) + O(n) + O(\max\{n, m\}) + \\ + O(\max\{n, m\}) = O(\max\{n, m\}) \quad (n, m \rightarrow \infty).$$

А так как

$$v_A(\mathbf{R}, n, m) \leq c \cdot v(\mathbf{R}, n, m),$$

где c – положительная константа, то

$$V_A(\mathbf{R}, n, m) = O(V(\mathbf{R}, n, m)) \quad (n, m \rightarrow \infty),$$

т.е. проверка того, являются ли изоморфными заданные графы $G_1, G_2 \in \mathbf{G}(n, m)$, осуществима с помощью алгоритма, локального для представления \mathbf{R} , что и требовалось показать.

Пример 1.10. Выберем в качестве исходных данных одноголовочной двумерной МТ с одной лентой типа \mathbf{Z}^2 представление $\mathbf{R}(G)$ графа $G \in \mathbf{G}(n, m)$ ($n \geq 3$) матрицей смежности (вершин или ребер)

$$\mathbf{M}_G = \begin{pmatrix} a_{11} & \cdots & a_{1k} \\ \vdots & \ddots & \vdots \\ a_{k1} & \cdots & a_{kk} \end{pmatrix},$$

где, соответственно, $k = n$ для матрицы смежности вершин и $k = m$ для матрицы смежности ребер. Тогда (см. пример 1.6)

$$V(\mathbf{R}, n, m) = O(k^2) \quad (k \rightarrow \infty).$$

Пусть $\mathbf{A} = (\alpha_{ij})$ и $\mathbf{B} = (\beta_{ij})$ – булевы матрицы k -го порядка. Обозначим через $\mathbf{A} \otimes \mathbf{B}$ такую булеву матрицу $\mathbf{C} = (\gamma_{ij})$ k -го порядка, что

$$\gamma_{ij} = \bigvee_{h=1}^k (\alpha_{ih} \wedge \beta_{hj}) \quad (i, j = 1, \dots, k).$$

Положим $\mathbf{A}^1 = \mathbf{A}$ и $\mathbf{A}^{\otimes l} = \underbrace{\mathbf{A} \otimes \dots \otimes \mathbf{A}}_{l \text{ раз}} \quad (l \geq 2)$.

I. Покажем, что проверка графа $G \in \mathbf{G}(n, m)$ на наличие подграфов, являющихся треугольниками, осуществима с помощью алгоритма, локального для представления графа G как матрицей смежности вершин, так и матрицей смежности ребер при любом (допустимом) законе роста $m \rightarrow \infty$.

Рассмотрим матрицу

$$\mathbf{M}_G^{\otimes 3} = (\gamma_{ij}).$$

Из определения операции \otimes непосредственно вытекает, что:

1. Если $\mathbf{R}(G)$ – представление графа $G \in \mathbf{G}(n, m)$ матрицей смежности вершин, то $\gamma_{ii} = 1$ ($i \in \{1, \dots, n\}$) тогда и только тогда, когда в графе G существует $(i-i)$ -прогулка длины 3. Отметим, что $(i-i)$ -прогулка длины 3 есть не что иное, как цикл длины 3, т.е. треугольник, содержащий вершину i .

2. Если $\mathbf{R}(G)$ – представление графа $G \in \mathbf{G}(n, m)$ матрицей смежности ребер, то $\gamma_{ii} = 1$ ($i \in \{1, \dots, m\}$) тогда и только тогда, когда в графе G существует прогулка длины 4, начинающаяся и заканчивающаяся ребром i . Отметим, что прогулка длины 4, начинающаяся и заканчивающаяся ребром i , есть не что иное, как прогулка по циклу длины 3, т.е. по треугольнику, содержащему ребро i .

Таким образом, для проверки графа $G \in \mathbf{G}(n, m)$ на наличие подграфов, являющихся треугольниками, достаточно вычислить матрицу $\mathbf{M}_G^{\otimes 3} = (\gamma_{ij})$, что осуществимо в рамках памяти объема $O(k^2)$ ($k \rightarrow \infty$). Если существует такое $i \in \{1, \dots, k\}$, что $\gamma_{ii} = 1$, то граф G содержит подграф, являющийся треугольником. Если же $\gamma_{ii} = 0$ для всех $i \in \{1, \dots, k\}$, то ни один подграф графа G не является треугольником. Для предложенного алгоритма A истинно неравенство

$$v_A(\mathbf{R}, n, m) \leq c \cdot v(\mathbf{R}, n, m),$$

где c – положительная константа. Следовательно,

$$V_A(\mathbf{R}, n, m) = O(V(\mathbf{R}, n, m)) \quad (n, m \rightarrow \infty),$$

т.е. проверка графа $G \in \mathbf{G}(n, m)$ на наличие подграфов, являющихся треугольниками, осуществима с помощью алгоритма, локального для представления графа G как матрицей смежности вершин, так и матрицей смежности ребер, что и требовалось показать.

II. Покажем, что проверка для графа $G \in \mathbf{G}(n, m)$ свойства «*быть двудольным графом*» осуществима с помощью алгоритма, локального для представления \mathbf{R} графа G матрицей смежности вершин \mathbf{M}_G при любом (допустимом) законе роста $m \rightarrow \infty$.

В соответствии с известным критерием (см. теорему 1.10), граф $G \in \mathbf{G}(n, m)$ – двудольный тогда и только тогда, когда он не содержит циклов нечетной длины. Воспользуемся этим критерием.

Определим на множестве булевых матриц k -го порядка бинарные операции \vee и \wedge , а также унарную операцию $\bar{}$ следующим образом: если $\mathbf{A} = (\alpha_{ij})$ и $\mathbf{B} = (\beta_{ij})$ – булевы матрицы k -го порядка, то

$$\mathbf{A} \vee \mathbf{B} = (\alpha_{ij} \vee \beta_{ij}),$$

$$\mathbf{A} \wedge \mathbf{B} = (\alpha_{ij} \wedge \beta_{ij}),$$

$$\bar{\mathbf{A}} = (\bar{\alpha}_{ij}).$$

Отметим, что определенные выше операции удовлетворяют законам де Моргана, т.е.

$$\overline{\mathbf{A} \vee \mathbf{B}} = \bar{\mathbf{A}} \wedge \bar{\mathbf{B}},$$

$$\overline{\mathbf{A} \wedge \mathbf{B}} = \bar{\mathbf{A}} \vee \bar{\mathbf{B}}.$$

Рассмотрим последовательность матриц

$$\mathbf{M}_G(1), \mathbf{M}_G(2), \dots, \mathbf{M}_G(n),$$

n -го порядка, определяемую рекуррентным соотношением

$$\mathbf{M}_G(l+1) = (\mathbf{M}_G(l) \otimes \mathbf{M}_G) \wedge \mathbf{K}_G(l) \quad (l = 2, \dots, n-1),$$

и начальными условиями

$$\mathbf{M}_G(1) = \mathbf{M}_G,$$

$$\mathbf{M}_G(2) = \mathbf{M}_G^{\otimes 2} \wedge \bar{\mathbf{I}}_n \wedge \bar{\mathbf{M}}_G,$$

где \mathbf{I}_n – единичная матрица n -го порядка, а матрицы $\mathbf{K}_G(l)$ ($l = 2, \dots, n-1$) определяются рекуррентным соотношением

$$\mathbf{K}_G(h+1) = \mathbf{K}_G(h) \wedge \overline{\mathbf{M}_G(h+1)} \quad (h = 2, \dots, n-2)$$

и начальным условием

$$\mathbf{K}_G(2) = \overline{\mathbf{M}_G(1)} \wedge \overline{\mathbf{M}_G(2)}.$$

Индукцией по числу r нетрудно показать, что каждая матрица $\mathbf{M}_G(r) = (\alpha_{ij}^{(r)})$ ($r = 1, \dots, n$) удовлетворяет условию:

$$\alpha_{ij}^{(r)} = 1 \quad (i \neq j; i, j = 1, \dots, n)$$

тогда и только тогда, когда длина кратчайшего $(i-j)$ -пути равна r , а

$$\alpha_{ii}^{(r)} = 1 \quad (i = 1, \dots, n)$$

тогда и только тогда, когда длина кратчайшего цикла, проходящего через вершину i , равна r .

Таким образом, проверка для графа $G \in \mathbf{G}(n, m)$ свойства «*быть двудольным графом*» сводится к последовательному вычислению матриц

$$\mathbf{M}_G(2l-1) = (\alpha_{ij}^{(2l-1)}) \quad (l = 2, \dots, \lfloor 0.5 \cdot (n+1) \rfloor).$$

Если для некоторого значения $l \in \{2, \dots, \lfloor 0.5 \cdot (n+1) \rfloor\}$ существует такое значение $i \in \{1, \dots, n\}$, что

$$\alpha_{ii}^{(2l-1)} = 1,$$

то граф G содержит цикл длины $2l-1$. Если же

$$\alpha_{ii}^{(2l-1)} = 0$$

для всех $i \in \{1, \dots, n\}$ при всех $l = 2, \dots, \lfloor 0.5 \cdot (n+1) \rfloor$, то граф G не содержит циклов нечетной длины.

Вычисление матрицы $\mathbf{M}_G(l)$ ($l = 2, \dots, n$), как и любой матрицы, определяемой рекуррентным соотношением ограниченной длины, осуществимо в рамках памяти размера $O(n^2)$ ($n \rightarrow \infty$). Следовательно, емкостная сложность предложенного алгоритма A равна

$$V_A(\mathbf{R}, n, m) = O(n^2) \quad (n \rightarrow \infty).$$

А так как

$$v_A(\mathbf{R}, n, m) \leq c \cdot v(\mathbf{R}, n, m),$$

где c – положительная константа, то

$$V_A(\mathbf{R}, n, m) = O(V(\mathbf{R}, n, m)) \quad (n, m \rightarrow \infty),$$

т.е. проверка для графа $G \in \mathbf{G}(n, m)$ свойства «*быть двудольным графом*» осуществима с помощью алгоритма, локального для представления \mathbf{R} графа $G \in \mathbf{G}(n, m)$ матрицей смежности вершин \mathbf{M}_G , что и требовалось показать.

III. Сфера применения последовательности матриц

$$\mathbf{M}_G(r) = (\alpha_{ij}^{(r)}) \quad (r = 1, \dots, n)$$

значительно превосходит поставленную выше цель, а именно: доказательство осуществимости проверки свойства «*быть двудольным графом*» с помощью алгоритма, локального для представления \mathbf{R} графа $G \in \mathbf{G}(n, m)$ матрицей смежности вершин \mathbf{M}_G . Эта сфера применения характеризуется унифицированным доказательством возможности решения исследуемой задачи с помощью алгоритма, локального для представления \mathbf{R} графа $G \in \mathbf{G}(n, m)$ матрицей смежности вершин \mathbf{M}_G , на основе условия: $\alpha_{ij}^{(r)} = 1$ ($i \neq j$; $i, j = 1, \dots, n$) тогда и только тогда, когда длина кратчайшего $(i - j)$ -пути равна r , а $\alpha_{ii}^{(r)} = 1$ ($i = 1, \dots, n$) тогда и только тогда, когда длина кратчайшего цикла, содержащего вершину i , равна r . Действительно, определим последовательность матриц

$$\mathbf{L}_G(1), \mathbf{L}_G(2), \dots, \mathbf{L}_G(n),$$

рекуррентным соотношением

$$\mathbf{L}_G(r+1) = \mathbf{L}_G(r) \vee \mathbf{M}_G(r+1) \quad (r = 1, \dots, n-1)$$

и начальным условием

$$\mathbf{L}_G(1) = \mathbf{M}_G.$$

Так как

$$\mathbf{L}_G(r) = \overline{\mathbf{K}_G(r)} \quad (r = 2, \dots, n),$$

где

$$\mathbf{K}_G(n) = \mathbf{K}_G(n-1) \wedge \overline{\mathbf{M}_G(n)},$$

то матрица

$$\mathbf{L}_G(r) = (\delta_{ij}^{(r)}) \quad (r = 1, \dots, n)$$

удовлетворяет условию:

$$\delta_{ij}^{(r)} = 1 \quad (i \neq j; i, j = 1, \dots, n)$$

тогда и только тогда, когда длина кратчайшего $(i - j)$ -пути не превосходит r , а

$$\delta_{ii}^{(r)} = 1 \quad (i = 1, \dots, n)$$

тогда и только тогда, когда длина кратчайшего цикла, содержащего вершину i , не больше, чем r .

Каждая матрица

$$\mathbf{L}_G(r) = (\delta_{ij}^{(r)}) \quad (r = 1, \dots, n)$$

может быть вычислена с помощью алгоритма, локального для представления \mathbf{R} графа $G \in \mathbf{G}(n, m)$ матрицей смежности вершин \mathbf{M}_G . Следовательно, в частности, с помощью алгоритма, локального для представления \mathbf{R} графа $G \in \mathbf{G}(n, m)$ матрицей смежности вершин \mathbf{M}_G может быть осуществлено решение следующих задач:

1. Проверка для графа $G \in \mathbf{G}(n, m)$ свойства «*быть связным графом*» при любом (допустимом) законе роста $m \rightarrow \infty$.

Действительно, граф $G \in \mathbf{G}(n, m)$ связан тогда и только тогда, когда существует такое $r \in \{1, \dots, n - 1\}$, что в каждой строке

$$(\delta_{i1}^{(r)}, \delta_{i2}^{(r)}, \dots, \delta_{in}^{(r)}) \quad (i = 1, \dots, n)$$

матрицы $\mathbf{L}(r)$ равны единице все компоненты, за исключением, быть может, компоненты, стоящей на главной диагонали, т.е.

$$(\exists r \in \{1, \dots, n\})(\forall i \in \{1, \dots, n\})(\forall j \in \{1, \dots, n\})(j \neq i \Rightarrow \delta_{ij}^{(r)} = 1).$$

Следовательно, проверка для графа $G \in \mathbf{G}(n, m)$ свойства «*быть связным графом*» осуществима с помощью алгоритма, локального для представления \mathbf{R} графа $G \in \mathbf{G}(n, m)$ матрицей смежности вершин \mathbf{M}_G .

2. Вычисление диаметра графа $G \in \mathbf{G}(n, m)$ при любом (допустимом) законе роста $m \rightarrow \infty$.

Пусть

$$r_0 = \min \{r \mid \mathbf{L}_G(r) \wedge \overline{\mathbf{I}}_n = \mathbf{L}_G(r + 1) \wedge \overline{\mathbf{I}}_n \}.$$

Если в каждой строке

$$(\delta_{i1}^{(r_0)}, \delta_{i2}^{(r_0)}, \dots, \delta_{in}^{(r_0)}) \quad (i = 1, \dots, n)$$

матрицы $\mathbf{L}(r_0)$ равны единице все компоненты, за исключением, быть может, компоненты, стоящей на главной диагонали, т.е.

$$(\forall i \in \{1, \dots, n\})(\forall j \in \{1, \dots, n\})(j \neq i \Rightarrow \delta_{ij}^{(r_0)} = 1),$$

то

$$\text{Diam } G = r_0,$$

иначе

$$\text{Diam } G = \infty.$$

Следовательно, вычисление диаметра графа $G \in \mathbf{G}(n, m)$ осуществимо с помощью алгоритма, локального для представления \mathbf{R} графа $G \in \mathbf{G}(n, m)$ матрицей смежности вершин \mathbf{M}_G .

3. Вычисление радиуса графа $G \in \mathbf{G}(n, m)$ при любом (допустимом) законе роста $m \rightarrow \infty$.

Для каждой вершины $i \in \{1, \dots, n\}$ графа $G \in \mathbf{G}(n, m)$ число

$$l_i = \max_{j \in \{1, \dots, n\}} d(i, j) \quad (i \in \{1, \dots, n\})$$

вычисляется следующим образом: l_i представляет собой такое наименьшее число r_i , что i -е строки матриц $\mathbf{L}(r_i)$ и $\mathbf{L}(r_i + 1)$, соответственно,

$$(\delta_{i1}^{(r_i)}, \delta_{i2}^{(r_i)}, \dots, \delta_{in}^{(r_i)})$$

и

$$(\delta_{i1}^{(r_i+1)}, \delta_{i2}^{(r_i+1)}, \dots, \delta_{im}^{(r_i+1)}),$$

совпадают с точностью до i -й компоненты, т.е.

$$(\forall j \in \{1, \dots, n\})(j \neq i \Rightarrow \delta_{ij}^{(r_i)} = \delta_{ij}^{(r_i+1)}),$$

причем все компоненты этих строк, не стоящие на главной диагонали, равны единице, т.е.

$$(\forall j \in \{1, \dots, n\})(j \neq i \Rightarrow \delta_{ij}^{(r_i)} = 1),$$

если такое число r_i существует, иначе

$$l_i = \infty.$$

Отметим, что элементы последовательности

$$l_1, \dots, l_n$$

удовлетворяют условию: либо все элементы последовательности – натуральные числа, удовлетворяющие неравенству

$$x \leq n - 1,$$

либо все элементы последовательности равны ∞ , т.е.

$$\infty \in \{l_i \mid i = 1, \dots, n\} \Rightarrow (\forall i \in \{1, \dots, n\})(l_i = \infty).$$

Таким образом, для вычисления радиуса графа $G \in \mathbf{G}(n, m)$ достаточно вычислить последовательность

$$l_1, \dots, l_n,$$

что осуществимо с помощью алгоритма, локального для представления \mathbf{R} графа $G \in \mathbf{G}(n, m)$ матрицей смежности вершин \mathbf{M}_G , а затем вычислить

$$\text{rad } G = \min \{l_i \mid i = 1, \dots, n\},$$

что также осуществимо с помощью алгоритма, локального для представления \mathbf{R} графа $G \in \mathbf{G}(n, m)$ матрицей смежности вершин \mathbf{M}_G .

4. Вычисление центра графа $G \in \mathbf{G}(n, m)$ при любом (допустимом) законе роста $m \rightarrow \infty$.

Так как

$$\text{cntr } G = \{i \mid \text{rad } G = l_i\},$$

то вычисление центра графа $G \in \mathbf{G}(n, m)$ осуществимо с помощью алгоритма, локального для представления \mathbf{R} графа $G \in \mathbf{G}(n, m)$ матрицей смежности вершин \mathbf{M}_G .

5. Выделение компонент связности графа $G \in \mathbf{G}(n, m)$ при любом (допустимом) законе роста $m \rightarrow \infty$.

Для каждой вершины $i \in \{1, \dots, n\}$ графа $G \in \mathbf{G}(n, m)$ обозначим через h_i такое наименьшее число, что i -е строки матриц $\mathbf{L}(h_i)$ и $\mathbf{L}(h_i + 1)$, соответственно,

$$(\delta_{i1}^{(h_i)}, \delta_{i2}^{(h_i)}, \dots, \delta_{in}^{(h_i)})$$

и

$$(\delta_{i1}^{(h_i+1)}, \delta_{i2}^{(h_i+1)}, \dots, \delta_{in}^{(h_i+1)}),$$

совпадают с точностью до i -й компоненты, т.е.

$$(\forall j \in \{1, \dots, n\})(j \neq i \Rightarrow \delta_{ij}^{(h_i)} = \delta_{ij}^{(h_i+1)}).$$

Ясно, что множество вершин

$$V_i = \{i\} \cup \{j \mid \delta_{ij}^{(h_i)} = 1\} \quad (i \in \{1, \dots, n\})$$

определяет компоненту связности G_i ($i \in \{1, \dots, n\}$) графа $G \in \mathbf{G}(n, m)$, содержащую вершину i .

Для построения компоненты связности G_i ($i \in \{1, \dots, n\}$) в явном виде достаточно «вырезать» те столбцы и строки матрицы смежности вершин \mathbf{M}_G , номера которых принадлежат множеству V_i .

Таким образом, выделение компонент связности графа $G \in \mathbf{G}(n, m)$ осуществимо с помощью алгоритма, локального для представления \mathbf{R} графа $G \in \mathbf{G}(n, m)$ матрицей смежности вершин \mathbf{M}_G .

Обозначим через $Lcl(\mathbf{R})$ класс задач теории графов, для которых существуют алгоритмы, локальные при выборе соответствующего представления \mathbf{R} графов $G \in \mathbf{G}(n, m)$ ($n, m \rightarrow \infty$). Именно вопрос о принадлежности задачи теории графов классу $Lcl(\mathbf{R})$ (при выборе соответствующего представления \mathbf{R} графов $G \in \mathbf{G}(n, m)$) является основным предметом исследования в настоящей монографии. В соответствии со стандартным подходом в теории алгоритмов (см., напр. [1,3,4,9]) этот класс задач, в общих чертах, может быть охарактеризован следующим образом (см. рис. 1.13). В классе $Prblms(\mathbf{R})$ всех задач теории графов, решаемых в терминах представления \mathbf{R} графов $G \in \mathbf{G}(n, m)$, естественно выделяется класс задач $P-SPACE(\mathbf{R})$, для которых существуют алгоритмы, *полиномиальные* по емкостной сложности. Класс $Lcl(\mathbf{R})$ является подклассом класса $P-SPACE(\mathbf{R})$, состоящим из всех задач, для которых существуют алгоритмы, емкостная сложность которых – *линейная функция* (т.е. полином 1-й степени) от размера входа.

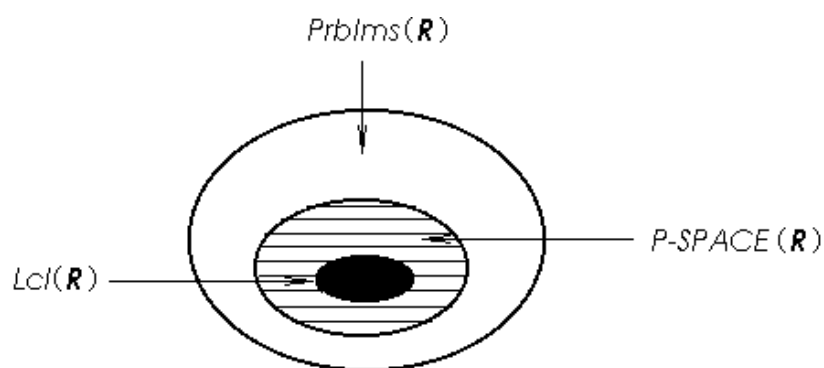


Рис. 1.13. Гипотетическая картина класса $Lcl(\mathbf{R})$.

Замечание 1.3. Класс $P-SPACE(\mathbf{R})$ может быть охарактеризован как класс языков, формулируемых в терминах представления $\mathbf{R}(G)$, для распознавания которых существуют алгоритмы, полиномиальные по емкостной сложности. С этой точки зре-

ния класс $LCI(\mathbf{R})$ представляет собой класс языков, формулируемых в терминах представления $\mathbf{R}(G)$ и распознаваемых с линейной памятью.

Из рассмотренных выше примеров вытекает, что класс $LCI(\mathbf{R})$ является достаточно представительным. Следующий пример показывает, что существуют также и задачи теории графов, для решения которых при выбранной модели вычислений отсутствуют локальные алгоритмы при любом представлении $\mathbf{R}(G)$ графа $G \in \mathbf{G}(n, m)$.

Пример 1.11. Выберем в качестве модели вычислений одноголовочную двумерную МТ с одной лентой типа \mathbf{Z}^2 .

Рассмотрим следующую задачу: по заданному представлению $\mathbf{R}(G)$ графа $G \in \mathbf{G}(n, m)$ построить представление $\mathbf{R}(L(G))$ реберного графа $L(G)$.

В п.1.2 было отмечено, что $L(G)$ является $(m, 0.5 \cdot \sum_{i=1}^n d_i^2 - m)$ -графом.

1. Пусть неограниченный рост n и m осуществляется так, что

$$n = o(m) \quad (m \rightarrow \infty). \quad (**)$$

а в качестве исходных данных одноголовочной двумерной МТ с одной лентой типа \mathbf{Z}^2 выбрано представление $\mathbf{R}(G)$ графа $G \in \mathbf{G}(n, m)$ матрицей смежности вершин

$$\mathbf{M}_G = \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{pmatrix}.$$

Тогда (см. пример 1.6)

$$V(\mathbf{R}, n, m) = O(n^2) \quad (n \rightarrow \infty).$$

Так как $\mathbf{M}_{L(G)}$ – матрица порядка $m \times m$, то для ее представления необходима память размера $c_1 \cdot m^2$, где c_1 – положительная константа. Следовательно, для любого построенного в рамках одноголовочной двумерной МТ с одной лентой типа \mathbf{Z}^2 алгоритма \mathbf{A} , осуществляющего построение матрицы $\mathbf{M}_{L(G)}$, справедливо неравенство

$$v_{\mathbf{A}}(\mathbf{R}, n, m) \geq c \cdot m^2, \quad (***)$$

где c – положительная константа. Из (**) и (***) вытекает, что

$$V(\mathbf{R}, n, m) = o(V_{\mathbf{A}}(\mathbf{R}, n, m)) \quad (n, m \rightarrow \infty).$$

Следовательно,

$$V_{\mathbf{A}}(\mathbf{R}, n, m) \neq O(V(\mathbf{R}, n, m)) \quad (n, m \rightarrow \infty),$$

т.е. существует такой закон роста $m \rightarrow \infty$, что алгоритм A не является локальным для представления \mathbf{R} .

2. Пусть в качестве исходных данных одноголовочной двумерной МТ с одной лентой типа \mathbf{Z}^2 выбрано представление $\mathbf{R}(G)$ графа $G \in \mathbf{G}(n, m)$ списками смежности вершин.

Тогда (см. пример 1.6)

$$V(\mathbf{R}, n, m) = O(m \cdot \max\{1, \min\{n, m\}\}) \quad (n, m \rightarrow \infty).$$

При $m = \binom{n}{2}$, т.е. для последовательности полных графов K_n ($n \in \mathbf{N}$), получим

$$V(\mathbf{R}, n, m) = O(n^2) \quad (n \rightarrow \infty). \quad (+)$$

Граф $L(K_n)$ является $\left(\binom{n}{2}, \binom{n}{2} \cdot (n-2)\right)$ -графом. Следовательно, для любого построенного в рамках одноголовочной двумерной МТ с одной лентой типа \mathbf{Z}^2 алгоритма A , осуществляющего построение списков смежности графа $L(K_n)$, справедливо неравенство

$$v_A(\mathbf{R}, n, m) \geq c \cdot n^3, \quad (++)$$

где c – положительная константа. Из соотношений (+) и (++) вытекает, что если $m = \binom{n}{2}$, то

$$V(\mathbf{R}, n, m) = o(V_A(\mathbf{R}, n, m)) \quad (n, m \rightarrow \infty).$$

Следовательно,

$$V_A(\mathbf{R}, n, m) \neq O(V(\mathbf{R}, n, m)) \quad (n, m \rightarrow \infty),$$

т.е. существует такой закон роста $m \rightarrow \infty$, что алгоритм A не является локальным для представления \mathbf{R} .

Особенностью задачи, рассмотренной в примере 1.11, является то, что размер *выхода*, по порядку, больше, чем размер *входа*. Поэтому, с содержательной точки зрения, нетривиальным и естественным является случай, когда в качестве модели вычислений выбрана МТ с выходной лентой. Такая модель дает возможность выделить в классе $Prblms(\mathbf{R})$ класс $P-WORK-SPACE(\mathbf{R})$ задач теории графов, для которых существует алгоритм с полиномиальной рабочей памятью. В свою очередь, в классе $P-WORK-SPACE(\mathbf{R})$ естественно может быть выделен класс

$WORK - Lcl(\mathbf{R})$ задач теории графов, для которых существует алгоритм, с рабочей памятью, линейной от размера представления \mathbf{R} . Непосредственно из определения введенных классов задач вытекает, что истинными являются включения

$$P - SPACE(\mathbf{R}) \subseteq P - WORK - SPACE(\mathbf{R}),$$

$$Lcl(\mathbf{R}) \subseteq WORK - Lcl(\mathbf{R}).$$

С учетом введенных выше классов задач, уточненная картина для класса $Lcl(\mathbf{R})$ показана на рис. 1.14.

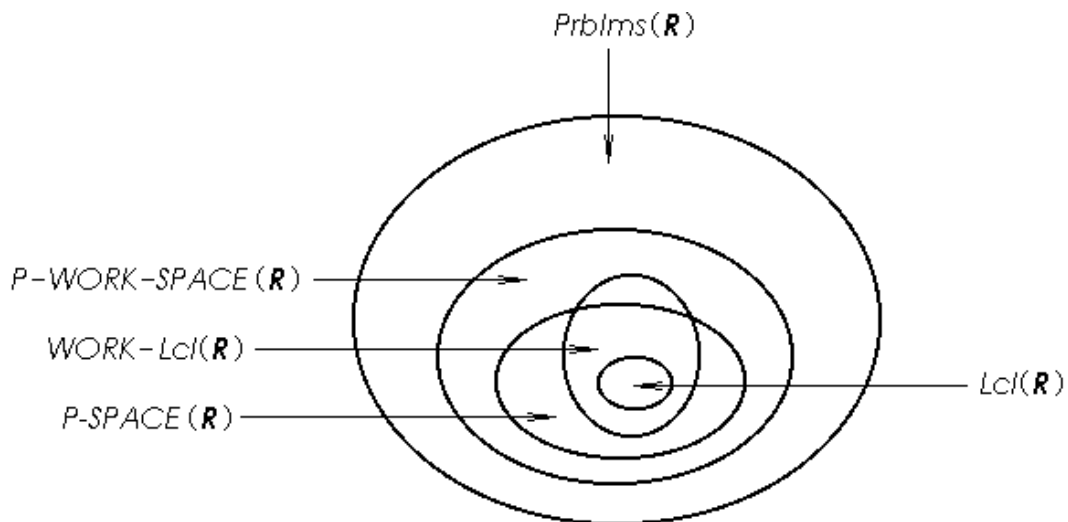


Рис.1.14. Уточненная гипотетическая картина класса $Lcl(\mathbf{R})$.

Как показывает следующая теорема, классификация, представленная на рис.1.14, действительно является более тонкой, чем классификация, представленная на рис.1.13.

Теорема 1.26. Если \mathbf{R} представление графа $G \in \mathbf{G}(n, m)$ либо матрицей, либо списками, смежности вершин, то существует такой закон роста $m \rightarrow \infty$, что

$$Lcl(\mathbf{R}) \subset WORK - Lcl(\mathbf{R}) \quad (1.1).$$

Доказательство. Пусть \mathbf{R} – представление графа $G \in \mathbf{G}(n, m)$ либо матрицей, либо списками, смежности вершин.

Рассмотрим задачу P_{LG} построения представления $\mathbf{R}(L(G))$ реберного графа $L(G)$ по представлению $\mathbf{R}(G)$ графа $G \in \mathbf{G}(n, m)$.

В примере 1.11 было показано, что существуют такие законы роста $m \rightarrow \infty$, что

$$P_{LG} \notin Lcl(\mathbf{R}) \quad (1.2)$$

для представления графа $G \in \mathbf{G}(n, m)$ как матрицей, так и списками, смежности вершин.

Покажем, что для обоих этих представлений (т.е. как матрицей, так и списками, смежности вершин) графа $G \in \mathbf{G}(n, m)$

$$P_{LG} \in WORK - Lcl(\mathbf{R}) \quad (1.3)$$

при любом (допустимом) законе роста $m \rightarrow \infty$.

Первая задача, которая возникает в процессе построения представления $\mathbf{R}(L(G))$ реберного графа $L(G)$ по представлению $\mathbf{R}(G)$ графа $G \in \mathbf{G}(n, m)$ либо матрицей, либо списками, смежности вершин, состоит в нумерации ребер графа $L(G)$ элементами множества $\{1, \dots, m\}$.

Замечание 1.4. Определение класса $Lcl(\mathbf{R})$ включает в себя условие $n, m \rightarrow \infty$, так что $G \neq \bar{K}_n$, т.е. $m \neq 0$. Для графа $G = \bar{K}_n$ матрица смежности вершин – нулевая, а каждый из списков смежности вершин – пустой. Проверка этих свойств осуществима с емкостной сложностью $V(\mathbf{R}, n, m)$ и в случае их выполнения на выходной ленте МТ выдается символ, являющийся кодом сообщения «граф $L(G)$ не содержит вершин» и МТ останавливается. Поэтому в дальнейшем предполагается, что $m \rightarrow \infty$, причем $m \neq 0$.

Любое отображение

$$f_{\text{нум}} : E \rightarrow \{1, \dots, m\} \quad (E \subseteq V^{(2)}),$$

осуществляющее нумерацию ребер графа $G \in \mathbf{G}(n, m)$, а, следовательно, и вершин графа $L(G)$, может быть представлено в виде следующей таблицы

Таблица 1.1.

1	2	...	m
$\{i_1, j_1\}$	$\{i_2, j_2\}$...	$\{i_m, j_m\}$

где

$$E = \{\{i_1, j_1\}, \{i_2, j_2\}, \dots, \{i_m, j_m\}\}.$$

Объем памяти, необходимой для хранения таблицы 1.1, равен

$$V_{\text{табл}} = O(m) \quad (n, m \rightarrow \infty).$$

При этом для представления графа $G \in \mathbf{G}(n, m)$ как матрицей, так и списками, смежности вершин истинно равенство

$$V_{\text{табл}} = O(V(\mathbf{R}, n, m)) \quad (n, m \rightarrow \infty).$$

Для доказательства соотношения (1.3) нам понадобится следующая лемма.

Лемма 1.2. Задача построения таблицы 1.1, определяющей одно (безразлично, какое именно) отображение $f_{\text{нум}}$, осуществляющее нумерацию ребер графа $G \in \mathbf{G}(n, m)$, принадлежит классу $LCI(\mathbf{R})$.

Доказательство. Покажем вначале справедливость следующих двух утверждений.

Утверждение 1.1. Пусть \mathbf{R} – представление графа $G \in \mathbf{G}(n, m)$ либо матрицей, либо списками, смежности вершин. Задача определения по паре вершин $\{i_0, j_0\}$ графа G истинности утверждения

$$\{i_0, j_0\} \in E?$$

и, в случае утвердительного ответа, вычисления номера ребра $\{i_0, j_0\}$, принадлежит классу $LCI(\mathbf{R})$.

Доказательство. Рассмотрим по отдельности указанные в формулировке утверждения представления графа $G \in \mathbf{G}(n, m)$, построив в каждом случае соответствующий алгоритм $A_{\text{нум}}$, решающий сформулированную задачу с емкостной сложностью

$$V_{A_{\text{нум}}} = V(\mathbf{R}, n, m) \quad (n, m \rightarrow \infty).$$

1. Пусть \mathbf{R} – представление графа $G \in \mathbf{G}(n, m)$ матрицей смежности вершин

$$\mathbf{M}_G = \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{pmatrix}.$$

Выберем в качестве $A_{\text{нум}}^{\text{mtr}}$ следующий алгоритм ($CNTR$ – переменная, значение которой – номер ребра $\{i_0, j_0\}$ графа G , если такое ребро существует и 0, если пара $\{i_0, j_0\}$ не является ребром графа G)

Алгоритм $A_{\text{нум}}^{\text{mtr}}$.

Шаг 1. Если $i_0 < j_0$, то переход к шагу 4, иначе – к шагу 2.

Шаг 2. Если $i_0 = j_0$, то переход к шагу 15, иначе – к шагу 3

Шаг 3. $x := j_0$, $y := i_0$, $i_0 = x$, $j_0 := y$ и переход к шагу 4.

Шаг 4. $i := 1$, $j := i + 1$, $CNTR := 0$.

Шаг 5. Если $i < i_0$, то переход к шагу 6, иначе – к шагу 10.

Шаг 6. Если $a_{ij} = 1$, то $CNTR := CNTR + 1$.

Шаг 7. $j := j + 1$.

Шаг 8. Если $j \leq n$, то переход к шагу 6, иначе – к шагу 9.

Шаг 9. $i := i + 1$, $j := i + 1$ и переход к шагу 5.

Шаг 10. Если $a_{ij} = 1$, то $CNTR := CNTR + 1$.

Шаг 11. Если $j < j_0$, то переход к шагу 12, иначе – к шагу 13.

Шаг 12. $j := j + 1$ и переход к шагу 10.

Шаг 13. Если $a_{ij} = 1$, то переход к шагу 14, иначе – к шагу 15.

Шаг 14. Выдать значение $CNTR$ и *конец*.

Шаг 15. $CNTR := 0$. Выдать значение $CNTR$ и *конец*.

Очевидно, что алгоритм $A_{\text{нум}}^{\text{mtr}}$ решает сформулированную в утверждении 1.1 задачу. Емкостная сложность алгоритма $A_{\text{нум}}^{\text{mtr}}$ равна

$$V_{A_{\text{нум}}^{\text{mtr}}} = V(\mathbf{R}, n, m) \quad (n, m \rightarrow \infty),$$

что и требовалось показать.

2. Пусть \mathbf{R} – представление графа $G \in \mathbf{G}(n, m)$ списками смежности вершин. Без ограничения общности можно считать, что \mathbf{R} представляет собой такую n -элементную последовательность списков

$$\mathbf{LIST} : \mathbf{LIST}_1, \mathbf{LIST}_2, \dots, \mathbf{LIST}_n,$$

что элементы каждого списка \mathbf{LIST}_i ($i = 1, \dots, n$), т.е. вершины графа $G \in \mathbf{G}(n, m)$, смежные с вершиной i , расположены в порядке их возрастания.

Определим для списков, являющимися элементами последовательности **LIST**, следующие операции

$$EMPTY(LIST_i) = \begin{cases} 1, & \text{если список } LIST_i \text{ пуст} \\ 0, & \text{иначе} \end{cases} \quad (i = 1, \dots, n),$$

$$FIRST(LIST_i) = \begin{cases} 0, & \text{если список } LIST_i \text{ пуст} \\ \text{первый элемент списка } LIST_i, & \text{иначе} \end{cases} \quad (i = 1, \dots, n),$$

$$LAST(LIST_i, j) = \begin{cases} 1, & \text{если } j \text{ - последний элемент списка } LIST_i \\ 0, & \text{иначе} \end{cases} \quad (i = 1, \dots, n; j \in LIST_i),$$

$$NEXT(LIST_i, j) = \begin{cases} \text{элемент списка } LIST_i, \text{ следующий за элементом } j, \\ \text{если такой элемент существует} \\ 0, & \text{иначе} \end{cases} \quad (i = 1, \dots, n; j \in LIST_i).$$

Выберем в качестве $A_{\text{нум}}$ следующий алгоритм (*CNTR* – переменная, значение которой – номер ребра $\{i_0, j_0\}$ графа G , если такое ребро существует и 0, если пара $\{i_0, j_0\}$ не является ребром графа G)

Алгоритм $A_{\text{нум}}^{\text{lst}}$.

Шаг 1. Если $i_0 < j_0$, то переход к шагу 4, иначе – к шагу 2.

Шаг 2. Если $i_0 = j_0$, то переход к шагу 20, иначе – к шагу 3

Шаг 3. $x := j_0$, $y := i_0$, $i_0 := x$, $j_0 := y$ и переход к шагу 4.

Шаг 4. $i := 1$, $CNTR := 0$.

- Шаг 5.* Если $i < i_0$, то переход к шагу 6, иначе – к шагу 12.
- Шаг 6.* Если $EMPTY(LIST_i) = 1$, то переход к шагу 7, иначе переход к шагу 8.
- Шаг 7.* $i := i + 1$ и переход к шагу 5.
- Шаг 8.* $x := FIRST(LIST_i)$.
- Шаг 9.* $CNTR := CNTR + 1$.
- Шаг 10.* Если $LAST(LIST_i, x) = 1$, то переход к шагу 7, иначе переход к шагу 11.
- Шаг 11.* $x := NEXT(LIST_i, x)$ и переход к шагу 9.
- Шаг 12.* Если $EMPTY(LIST_i) = 1$, то переход к шагу 20, иначе переход к шагу 13.
- Шаг 13.* $x := FIRST(LIST_i)$.
- Шаг 14.* $CNTR := CNTR + 1$.
- Шаг 15.* Если $x < j_0$, то переход к шагу 16, иначе – к шагу 18.
- Шаг 16.* Если $LAST(LIST_i, x) = 1$, то переход к шагу 20, иначе переход к шагу 17.
- Шаг 17.* $x := NEXT(LIST_i, x)$ и переход к шагу 14.
- Шаг 18.* Если $x = j_0$, то переход к шагу 19, иначе – к шагу 20.
- Шаг 19.* Выдать значение $CNTR$ и *конец*.
- Шаг 20.* $CNTR := 0$. Выдать значение $CNTR$ и *конец*.

Очевидно, что алгоритм $A_{\text{нум}}^{\text{lst}}$ решает сформулированную в утверждении 1.1 задачу. Емкостная сложность алгоритма $A_{\text{нум}}^{\text{lst}}$ равна

$$V_{A_{\text{нум}}^{\text{lst}}} = V(\mathbf{R}, n, m) \quad (n, m \rightarrow \infty),$$

что и требовалось показать.

Утверждение доказано.

Утверждение 1.2. Пусть \mathbf{R} – представление графа $G \in \mathbf{G}(n, m)$ либо матрицей, либо списками, смежности вершин. Задача вычисления числа m ребер графа G , а, следовательно, числа вершин графа $L(G)$, принадлежит классу $LC(\mathbf{R})$.

Доказательство. Рассмотрим по отдельности указанные в формулировке утверждения представления графа $G \in \mathbf{G}(n, m)$, построив в каждом случае соответствующий алгоритм $A_{\text{edge-nubr}}$, решающий сформулированную задачу с емкостной сложностью

$$V_{A_{edge-nbr}} = V(\mathbf{R}, n, m) \quad (n, m \rightarrow \infty).$$

Для компактности записи представим алгоритм $A_{edge-nbr}$ на Упрощенном Алголе (см., напр. [1]).

1. Пусть \mathbf{R} – представление графа $G \in \mathbf{G}(n, m)$ матрицей смежности вершин.

$$\mathbf{M}_G = \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{pmatrix}.$$

Выберем в качестве $A_{edge-nbr}$ алгоритм $A_{edge-nbr}^{mtr}$, реализованный в виде следующей процедуры

Procedure EDGE_NMBR_MTR(\mathbf{M}_G).

```

begin
   $m := 0$ 
  do  $i = 1, \dots, n - 1$ 
    do  $j = i + 1, \dots, n$ 
      if  $a_{ij} = 1$ 
        then  $m := m + 1$ 
      end
    end
  end
end

```

Очевидно, что алгоритм $A_{edge-nbr}^{mtr}$ решает сформулированную в утверждении 1.2 задачу. Емкостная сложность алгоритма $A_{edge-nbr}^{mtr}$ равна

$$V_{A_{edge-nbr}^{mtr}} = V(\mathbf{R}, n, m) \quad (n, m \rightarrow \infty),$$

что и требовалось показать.

2. Пусть \mathbf{R} – представление графа $G \in \mathbf{G}(n, m)$ списками смежности вершин. Как и при доказательстве утверждения 1.1 считаем, что \mathbf{R} представляет собой такую n -элементную последовательность списков

$$\mathbf{LIST} : LIST_1, LIST_2, \dots, LIST_n,$$

что элементы каждого списка $LIST_i$ ($i = 1, \dots, n$) расположены в порядке их возрастания.

Выберем в качестве $A_{edge-nmbr}$ алгоритм $A_{edge-nmbr}^{lst}$, реализованный в виде следующей процедуры

Procedure EDGE_NMBR_LST(**LIST**).

```

begin
   $m := 0$ 
  do  $i = 1, \dots, n - 1$ 
    if  $EMPTY(LIST_i) = 1$ 
      then  $i := i + 1$ 
    else
      do
         $x := FIRST(LIST_i)$ 
        do while  $x \neq 0$ 
          if  $x > i$ 
            then  $m := m + 1$ 
             $x := NEXT(LIST_i, x)$ 
          end
        end
      end
    end
  end

```

Очевидно, что алгоритм $A_{edge-nmbr}^{lst}$ решает сформулированную в утверждении 1.2 задачу. Емкостная сложность алгоритма $A_{edge-nmbr}^{lst}$ равна

$$V_{A_{edge-nmbr}^{lst}} = V(\mathbf{R}, n, m) \quad (n, m \rightarrow \infty),$$

что и требовалось показать.

Утверждение доказано.

Перейдем теперь непосредственно к доказательству леммы 1.2.

Рассмотрим следующий алгоритм, осуществляющий построение таблицы 1.1. в два этапа.

На 1-м этапе вычислим число m ребер графа G (т.е. число вершин графа $L(G)$) с помощью алгоритма $A_{edge-nmbr}$, построенного в процессе до-

казательства утверждения 1.2, и заполним 1-ю строку таблицы 1.1. Емкостная сложность этих вычислений равна $V(\mathbf{R}, n, m)$ ($n, m \rightarrow \infty$).

На 2-м этапе заполним 2-ю строку таблицы 1.1. С этой целью применим к каждой паре вершин $\{i_0, j_0\}$ ($i_0 < j_0$) графа G алгоритм $A_{i_0 j_0}$, построенный в процессе доказательства утверждения 1.1. Этим самым мы выясняем, является ли пара $\{i_0, j_0\}$ ребром графа G (иными словами, вершиной графа $L(G)$) и, в случае положительного ответа, вычисляем номер этого ребра. В последнем случае записываем ребро $\{i_0, j_0\}$ в соответствующую клетку 2-й строки таблицы 1.1. Емкостная сложность этих вычислений равна $V(\mathbf{R}, n, m)$ ($n, m \rightarrow \infty$).

Предложенный алгоритм построения таблицы 1.1 имеет емкостную сложность, равную $V(\mathbf{R}, n, m)$ ($n, m \rightarrow \infty$). Следовательно, задача построения таблицы 1.1, определяющей одно (безразлично, какое именно) отображение $f_{i_0 j_0}$, осуществляющее нумерацию ребер графа $G \in \mathbf{G}(n, m)$, принадлежит классу $LCI(\mathbf{R})$, что и требовалось показать.

Лемма доказана.

Докажем теперь соотношение (1.3).

Для доказательства достаточно построить алгоритм A_{LG} , использующий таблицу 1.1 в качестве исходных данных и генерирующий представление $\mathbf{R}(L(G))$ графа $L(G)$ в рамках используемой на рабочих лентах МТ памяти, имеющей размер $V(\mathbf{R}, n, m)$ ($n, m \rightarrow \infty$). Построим такой алгоритм для каждого из рассматриваемых представлений графа $G \in \mathbf{G}(n, m)$ (т.е. либо матрицей, либо списками, смежности вершин) по отдельности.

1. Пусть \mathbf{R} – представление графа $G \in \mathbf{G}(n, m)$ матрицей смежности вершин. Выберем в качестве A_{LG} алгоритм A_{LG}^{mtr} , осуществляющий простую генерацию матрицы

$$\mathbf{M}_{L(G)} = \begin{pmatrix} \beta_{11} & \beta_{12} & \cdots & \beta_{1m} \\ \beta_{21} & \beta_{22} & \cdots & \beta_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ \beta_{m1} & \beta_{m2} & \cdots & \beta_{mm} \end{pmatrix}$$

смежности вершин графа $L(G)$ на выходную ленту МТ, реализованный в виде следующей процедуры

Procedure LINE_GRAPH_MTR(Таблица 1.1).

```

begin
  do  $k = 1, \dots, m$ 
    do  $h = 1, \dots, k - 1$ 
      if  $|\{i_h, j_h\} \cap \{i_k, j_k\}| = 1$ 
        then  $\gamma_h := 1$ 
        else  $\gamma_h := 0$ 
      end
       $\gamma_k := 0$ 
    do  $h = k + 1, \dots, m$ 
      if  $|\{i_h, j_h\} \cap \{i_k, j_k\}| = 1$ 
        then  $\gamma_h := 1$ 
        else  $\gamma_h := 0$ 
      end
    на выходной ленте МТ записать  $(\gamma_1, \dots, \gamma_m)$  в качестве  $k$ -й
    строки матрицы  $\mathbf{M}_{L(G)}$ 
  end
end

```

Так как

$$m = O(V(\mathbf{R}, n, m)) \quad (n, m \rightarrow \infty),$$

то объем памяти, необходимой для хранения булевого вектора

$$(\gamma_1, \dots, \gamma_m)$$

равен $V(\mathbf{R}, n, m)$ ($n, m \rightarrow \infty$). Следовательно, алгоритм A_{LG}^{mtr} осуществляет построчную генерацию матрицы $\mathbf{M}_{L(G)}$ смежности вершин графа $L(G)$ на выходную ленту МТ в рамках используемой на рабочих лентах МТ памяти, имеющей размер $V(\mathbf{R}, n, m)$ ($n, m \rightarrow \infty$), что и требовалось показать.

2. Пусть \mathbf{R} – представление графа $G \in \mathbf{G}(n, m)$ списками смежности вершин.

Определим операцию внесения элемента в список следующим образом. Пусть LST – список, элементы которого – натуральные числа (в частности, список LST может быть пустым). Тогда $INSRT(LST, x)$ ($x \in \mathbf{N}$) есть список, полученный в результате добавления элемента x в конец списка LST .

Выберем в качестве A_{LG} алгоритм A_{LG}^{lst} , осуществляющий последовательно генерацию элементов последовательности

$$LIST : LIST_1, LIST_2, \dots, LIST_m,$$

списков смежности вершин графа $L(G)$ на выходную ленту МТ, реализованный в виде следующей процедуры

Procedure LINE_GRAPH_LST(Таблица 1.1).

begin

do $k = 1, \dots, m$

$LIST_k := \emptyset$

do $h = 1, \dots, k - 1$

if $|\{i_h, j_h\} \cap \{i_k, j_k\}| = 1$

then $LIST_k := INSRT(LIST_k, h)$

end

do $h = k + 1, \dots, m$

if $|\{i_h, j_h\} \cap \{i_k, j_k\}| = 1$

then $LIST_k := INSRT(LIST_k, h)$

end

на выходной ленте МТ записать список $LIST_k$ в качестве k -го элемента последовательности **LIST**

end

end

Число элементов любого списка $LIST_k$ ($k = 1, \dots, m$) меньше, чем m .
А так как

$$m = O(V(\mathbf{R}, n, m)) \quad (n, m \rightarrow \infty),$$

то объем памяти, необходимой для хранения списка $LIST_k$ ($k = 1, \dots, m$) равен $V(\mathbf{R}, n, m)$ ($n, m \rightarrow \infty$). Следовательно, алгоритм A_{LG}^{lst} осуществляет последовательно генерацию списков смежности вершин графа $L(G)$ на выходную ленту МТ в рамках используемой на рабочих лентах МТ памяти, имеющей размер $V(\mathbf{R}, n, m)$ ($n, m \rightarrow \infty$), что и требовалось показать.

Соотношение (1.3) доказано.

Из (1.2) и (1.3) вытекает, что

$$Lcl(\mathbf{R}) \neq WORK - Lcl(\mathbf{R}).$$

Из этого соотношения и из включения

$$Lcl(\mathbf{R}) \subseteq WORK - Lcl(\mathbf{R}),$$

вытекает, что

$$Lcl(\mathbf{R}) \subset WORK - Lcl(\mathbf{R}),$$

что и требовалось показать.

Теорема доказана.

Замечание 1.5. Нетрудно заметить, что процедуры LINE_GRAPH_MTR и LINE_GRAPH_LST, используемые в процессе доказательства теоремы 1.26, легко преобразовать так, чтобы на выходной ленте МТ последовательно осуществлялась запись элементов, соответственно, строки матрицы смежности вершин $\mathbf{M}_{L(G)}$ и списка смежности вершин $LIST_k$ ($k = 1, \dots, m$), по мере их вычисления. В этом случае вообще отпадает необходимость хранения в рабочей памяти МТ, соответственно, вычисляемой строки матрицы смежности вершин $\mathbf{M}_{L(G)}$ или вычисляемого списка смежности вершин $LIST_k$ ($k = 1, \dots, m$). Указанное преобразование процедур LINE_GRAPH_MTR и LINE_GRAPH_LST повышает их эффективность (например, при их компьютерной реализации), но ничего не дает с точки зрения усиления теоремы 1.26. Именно по этой причине, а также чтобы не вносить излишние усложнения в процедуры, затеняющие их прозрачность, указанная модификация не была реализована.

Значение теоремы 1.26 состоит в следующем. Из включения (1.1) вытекает, что для представления \mathbf{R} графа $G \in \mathbf{G}(n, m)$ как матрицей смежности вершин, так и списками смежности вершин существуют такие законы роста $m \rightarrow \infty$, что истинно соотношение

$$WORK - Lcl(\mathbf{R}) \setminus Lcl(\mathbf{R}) \neq \emptyset.$$

Таким образом, в тех случаях, когда показано, что исследуемая задача теории графов не принадлежит классу $Lcl(\mathbf{R})$, естественным является вопрос о ее принадлежности классу $WORK - Lcl(\mathbf{R})$.

1.5. Выводы.

В течение последнего полувека наблюдается возникновение и постоянное углубление сотрудничества между теорией алгоритмов и теорией графов. Систематический интерес теории алгоритмов к задачам теории графов продемонстрирован в книгах [1,3,4,9], каждая из которых является *классическим шедевром* в области современных Computer Science. В свою очередь, систематический интерес теории графов к теории алгоритмов продемонстрирован в книгах [5,7], предмет исследования которых – конструирование и анализ алгоритмов на графах. Создание и развитие раздела, получившего имя *автоматы в лабиринтах* (см., напр., [6]), установило глубокие внутренние связи между теорией алгоритмов, теорией автоматов и теорией графов.

Традиционно в процессе развития теории алгоритмов превалировал *анализ временной сложности* решения исследуемой задачи. Именно этот аспект привел к созданию следующих двух разделов, определивших *лицо* современной теории алгоритмов. Первый из этих разделов посвящен *разработке техники конструирования эффективных алгоритмов*, т.е. алгоритмов, временная сложность которых – полином от размера входа. Именно этому разделу посвящены, в основном, книги [1,4,9]. Второй раздел получил имя *теория NP-полноты*. Объектами его исследования являются класс *NP* всех переборных задач и класс *P* переборных задач, разрешимых на МТ за полиномиальное время. Систематически этот раздел изложен в книге [3], а его основные положения представлены в каждой из книг [1,4,9]. *Анализ емкостной сложности* решения задач дискретной математики и ее многочисленных приложений, как правило, либо не проводился вообще, либо отходил на второй план. Сказанное в частности, в полной мере применимо и к теории графов.

Существенный рост быстродействия процессоров и появление сверхбольших интегральных схем памяти резко усилили возможности современных компьютеров. Как следствие, значительно расширился круг задач (вне зависимости от того, является ли задача переборной или нет), особенно из прикладных разделов Computer Science, решаемых с использованием компьютеров. В частности, получило широкое распространение применение мета эвристик направленного поиска для создания методов построения приближенных решений задач дискретной математики и ее многочисленных приложений (см., напр. [16]). В этой связи методы анализа емкостной сложности алгоритмов постепенно выходят на передний план. Поэтому исследование емкостной сложности алгоритмов в конкретных разделах дискретной математики становится актуальной проблемой, особенно в свете появления новых парадигм вычислений, таких как ДНК-вычисления (см., напр. [15]).

Теория графов является одним из основных разделов дискретной математики, малейшее продвижение в котором не раз оказывало существенное влияние как на дискретную математику, так и на ее многочисленные приложения. По этой причине систематический анализ емкостной сложности алгоритмов на графах имеет значение не только для теории графов, но и для всей дискретной математики, в частности, для теории алгоритмов.

Предметом исследования настоящей монографии является класс $Lcl(\mathbf{R})$ задач теории графов, для которых существуют алгоритмы, локальные для представления \mathbf{R} графа $G \in \mathbf{G}(n, m)$, т.е. алгоритмы, емкостная сложность которых является линейной функцией от размера входа. Рассмотренные в настоящем разделе многочисленные примеры показывают, что этот класс задач является довольно представительным и ему принадлежат многие задачи теории графов, играющие важную роль как в самой теории графов, так и в ее многочисленных приложениях. Из включения (1.1) вытекает, что если исследуемая задача $P \in Prblms(\mathbf{R})$ теории графов (решаемая в терминах представления \mathbf{R} графов $G \in \mathbf{G}(n, m)$) не принадлежит классу $Lcl(\mathbf{R})$, то естественно возникает вопрос о принадлежности задачи P классу $WORK - Lcl(\mathbf{R})$ задач теории графов, для которых существует алгоритм, с рабочей памятью, линейной от размера представления \mathbf{R} . Таким образом, класс $WORK - Lcl(\mathbf{R})$ играет для класса $Lcl(\mathbf{R})$ роль, в некотором смысле двойственную той роли, которую играет класс $DLOG - SPACE$ для класса P в теории *NP-полноты*.

2. АНАЛИЗ ПРЕДСТАВЛЕНИЙ ГРАФОВ.

Среди семи рассмотренных в предыдущем разделе графа (см. п. 1.2 и 1.4) представлений основную роль играют представления графа матрицами и списками.

Матрица является формальным математическим понятием, причем, с алгебраической точки зрения, одним из наиболее эффективных понятий. Соответствующая матрица для графа (т.е. матрица смежности вершин, матрица смежности ребер или матрица инцидентий) определяется единственным образом с точностью до перестановки строк и/или столбцов. Эта «неоднозначность» матричного представления, не оказывает никакого влияния на сложность операций, выполняемых над матрицами, так как перестановка, как строк, так и столбцов матрицы (которые, по своей сути, являются наиболее простыми операциями над матрицами) не увеличивает ни емкостной, ни временной сложности никаких других операций над матрицами.

Несколько иначе обстоит дело с представлениями графов совокупностями списков. «Свобода действий», связанная с неоднозначностью представления списками, существенно усложняет операции и над списками, и над их совокупностями. Действительно, как элементы списка, так и совокупность списков, могут быть упорядочены различными способами. Следовательно, как это обычно имеет место в процессе исследования математических структур, возникает необходимость в определении отношения эквивалентности \mathcal{E} , характеризующего отношение неразличимости объектов (т.е. списков и/или их совокупностей). Более того, возникает необходимость в построении процедуры, осуществляющей эффективное вычисление проверки принадлежности объектов одному и тому же классу отношения эквивалентности \mathcal{E} .

Описанная конструкция, по своей сути, с качественной точки зрения эквивалентна формальному представлению совокупности списков в виде алгебраической системы, содержащей системы множеств и определенных на них подстановок. Операции для такой алгебраической системы определяются в соответствии с орбитами, определяемыми теми или иными подгруппами симметрической группы. Как следствие, в результате указанной формализации возникают следующие два крайне неприятных обстоятельства. Во-первых, операции над представлениями графов в виде совокупности списков теряют свою прозрачность. Во-вторых, действия в соответствии с орбитами автоматически переводит представления графов совокупностями списков в класс матричных представлений, что, в конечном счете и определяет сложность выполняемых операций.

Наиболее простой и естественный способ устранения указанной выше сложности, связанной с неоднозначностью представления графа списками смежности, состоит в том, чтобы нивелировать эту неоднозначность с самого начала. Для решения этой задачи достаточно выделить «стандартную форму» представления графа списками смежности (такая форма, фактически, уже применялась в п.1.4 при доказательстве утверждения 1.1) и оперировать исключительно со стандартными формами.

Построение и анализ стандартных форм представления графов списками смежности и является основной целью настоящего раздела. В п.2.1 сделан ряд замечаний, связанных с различимостью представлений графов списками и матрицами смежности ребер. В п.2.2 понятие *стандартной формы представления списками смежности* (как вершин, так и ребер) рассмотрено для обычных графов. На основе этой формы выполнен сравнительный анализ эффективности представлений графов списками смежности и матрицами с позиции алгебраической системы на множестве графов. В п.2.3 развито обобщение понятия стандартной формы представления списками смежности для направленных графов и орграфов. П.2.4 содержит краткие выводы.

2.1. Предварительные замечания.

В п.1.4 было принято соглашение о том, что если $\mathbf{R}(G)$ – представление графа $G = (V, E) \in \mathbf{G}(n, m)$ ($n, m \in \mathbf{N}, 0 \leq m \leq \binom{n}{2}$) списками смежности ребер, матрицей смежности ребер или матрицей инцидентий, то

$$E = \{1, \dots, m\},$$

т.е. ребра графа G занумерованы натуральными числами. Отображение f , осуществляющее такую нумерацию ребер, в явном виде представлено в матрице инцидентий. Следующая теорема показывает, что полная информация об отображении f не содержится в списках смежности ребер. Следовательно, в силу эквивалентности представлений однотипными списками и матрицами, полная информация об отображении f не содержится и в матрице смежности ребер.

Теорема 2.1. При всех $n \geq 4$ и $m \geq 3$ существуют такие не изоморфные графы $G', G'' \in \mathbf{G}(n, m)$, что списки смежности их ребер идентичны друг другу с точностью до перестановок элементов списков.

Доказательство. Очевидно, что достаточно доказать теорему для случая, когда $n = 4$ и $m = 3$.

Рассмотрим графы $G', G'' \in \mathbf{G}(4, 3)$, изображенные на рис. 2.1.

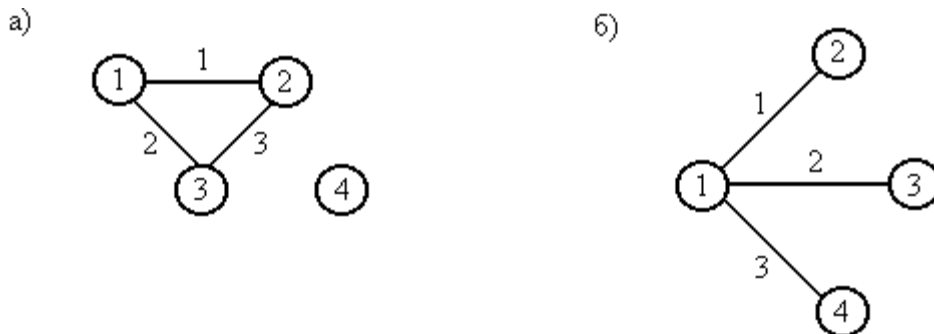


Рис. 2.1. Графическое изображение: а) графа G' ; б) графа G'' .

Графы G' и G'' – не изоморфные (хотя бы потому, что граф G' не является связным, в то время как G'' – связный граф). Списки смежности их ребер изображены на рис 2.2.

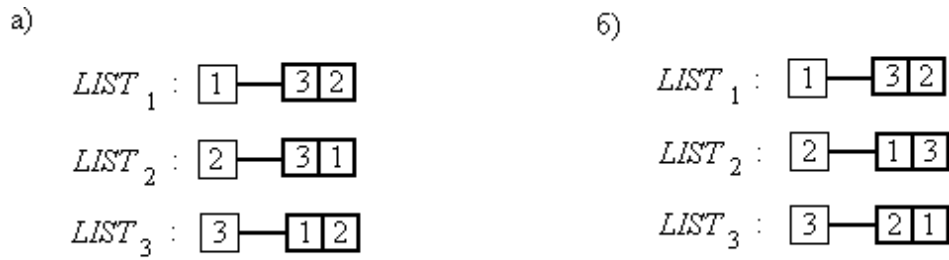


Рис. 2.2. Представление списками смежности ребер: а) графа G' ; б) графа G'' .

Сравнивая рис. 2.2. а) с рис. 2.2.б), заключаем, что, с точностью до перестановок элементов списков, списки смежности ребер графов G' и G'' идентичны друг другу, что и требовалось показать.

Теорема доказана.

Для того чтобы избежать ситуации, когда представления различных графов списками смежности ребер и (в силу эквивалентности представлений однотипными списками и матрицами) матрицами смежности ребер неразличимы, всюду в дальнейшем считаем, что кроме матрицы смежности ребер или, соответственно, списков смежности ребер, для каждого графа задана также и таблица 1.1. Отметим, что из леммы 1.2 вытекает, что как построение таблицы 1.1, так и работа с ней, не влияет на принадлежность любой из исследуемых задач классу $LCl(\mathbf{R})$.

Замечание 2.1. Неразличимость представлений неизоморфных графов, как списками смежности ребер, так и матрицами смежности ребер, можно было бы избежать и совершенно иным способом. Действительно, представим каждое ребро $\{i, j\}$ графа $G \in \mathbf{G}(n, m)$ ($n, m \in \mathbf{N}, 0 \leq m \leq \binom{n}{2}$) упорядоченной парой (i_0, j_0) , где $i_0 = \min\{i, j\}$, $j_0 = \max\{i, j\}$. Занумеруем ребро $\{i, j\}$ числом $l = i_0 + n \cdot j_0$. Зная число n вершин графа G , мы можем восстановить вершины, инцидентные ребру с номером номер l . Пусть выбран равномерный вес. Тогда емкостная сложность предложенного метода нумерации ребер графа $G \in \mathbf{G}(n, m)$ равна $O(1)$ ($n \rightarrow \infty$). Однако, при выбранном способе нумерации ребер, множество номеров ребер графа $G \in \mathbf{G}(n, m)$ представляет собой разреженное подмножество множества $\{2n + 1, \dots, n^2 + n - 1\}$. Это вызывает целый ряд неудобств при действиях, как со списками смежности ребер, так и с матрицами смежности ребер. Их нивелирование, в конечном итоге, приводит к принятому в настоящей работе способу нумерации ребер графа $G \in \mathbf{G}(n, m)$ при его представлении списками смежности ребер или матрицами смежности ребер.

2.2. Представления графов списками смежности.

Представления графа $G = (V, E) \in \mathbf{G}(n, m)$ ($n, m \in \mathbf{N}, 0 \leq m \leq \binom{n}{2}$), как списками смежности вершин, так и списками смежности ребер, имеют одну и ту же структуру. Поэтому, естественно определить стандартную форму этих представлений единым образом. Унифицированное определение имеет следующий вид.

Определение 2.1. Назовем *стандартной формой* представления графа $G = (V, E) \in \mathbf{G}(n, m)$ ($n, m \in \mathbf{N}, 0 \leq m \leq \binom{n}{2}$) списками смежности такую k -элементную последовательность списков

$$\mathbf{LIST} : LIST_1, LIST_2, \dots, LIST_k, \quad (2.1)$$

где

$$k = \begin{cases} n, & \text{для списков смежности вершин} \\ m, & \text{для списков смежности ребер} \end{cases}$$

что элементы каждого списка $LIST_i$ ($i = 1, \dots, k$) (т.е. вершины графа G , смежные с вершиной i , в случае списков смежности вершин, либо ребра графа G , смежные с ребром i , в случае списков смежности ребер) расположены в порядке их возрастания.

Замечание 2.2. 1. Исходя из операции $NEXT$ (см. п.1.4, доказательство утверждения 1.1), определим операции \mathbf{next}_r ($r = 1, \dots, k$) над списками, являющимися элементами последовательности (2.1), с помощью равенств

$$\mathbf{next}_r(j) = NEXT(LIST_r, j) \quad (r = 1, \dots, k)$$

и положим

$$\mathbf{next}_r^h = \underbrace{\mathbf{next}_r \circ \dots \circ \mathbf{next}_r}_{h \text{ раз}} \quad (r = 1, \dots, k; h \in \mathbf{N})$$

Из определения 2.1 вытекает, что стандартная форма представления графа $G = (V, E) \in \mathbf{G}(n, m)$ ($n, m \in \mathbf{N}, 0 \leq m \leq \binom{n}{2}$) списками смежности удовлетворяет следующему условию

$$(\forall r \in \{1, \dots, k\})(\forall i, j \in LIST_h)((\exists h \in \mathbf{N})(j = \mathbf{next}_r^h(i)) \Rightarrow i < j) \quad (2.2)$$

Из (2.2) непосредственно вытекает, что приведение любого представления \mathbf{R} графа $G \in \mathbf{G}(n, m)$ списками смежности к его стандартной форме естественно сводится к решению ряда *задач сортировки* (списков и/или множества списков) и, следова-

тельно, принадлежит классу $LC(\mathbf{R})$. Поэтому соглашение о стандартной форме представления графа $G \in \mathbf{G}(n, m)$ списками смежности упрощает изложение и не ограничивает общности рассуждений.

2. В определении 2.1 зафиксирован порядок, в котором элементы, принадлежащие линейно упорядоченному множеству $S = (\{1, \dots, k\}, <)$, расположены в каждом списке, принадлежащем последовательности (2.1). А именно, выбран порядок \prec , соответствующий возрастанию элементов в соответствии с их линейным порядком $<$ в исходном множестве, т.е. принято соглашение о том, что отношения порядка \prec и $<$ совпадают. Конкретный вид порядка, в котором расположены элементы в списке, вообще говоря, не является существенным. Однако для эффективной работы со списками необходимо, чтобы отношение порядка \prec можно было *легко вычислить* в терминах исходного отношения $<$. Иными словами, чтобы и временная, и емкостная сложность согласования порядков равнялась $O(1)$ ($k \rightarrow \infty$). В противном случае согласование порядков осуществляется, фактически, за счет применения фиксированной подстановки, определенной на исходном множестве элементов, т.е. имеет и временную, и емкостную сложность, равную $O(k)$ ($k \rightarrow \infty$). Поэтому, всюду в дальнейшем считаем, что отношение порядка \prec совпадает с отношением $<$. Это соглашение не влияет на общность рассуждений, а только упрощает изложение.

То, что при использовании списков важна именно фиксация порядка следования элементов, обусловлено следующим обстоятельством: представление подмножеств линейно упорядоченного множества списками с фиксированным порядком следования элементов является эффективной структурой данных, предназначенной для выполнения теоретико-множественных операций (см., напр. [1]). Действительно, пусть S – фиксированное конечное линейно упорядоченное множество. Рассмотрим множества S_1 и S_2 ($S_1, S_2 \subseteq S$), заданные списками, элементы которых расположены в порядке их возрастания. Тогда:

1) построение каждого из множеств $S_1 \cap S_2$, $S_1 \cup S_2$, $S_1 \setminus S_2$, а, следовательно, и множества $S_1 \oplus S_2$, осуществимо за время

$$T = O(\max\{|S_1|, |S_2|\}) \quad (|S_1|, |S_2| \rightarrow \infty);$$

2) проверку, как включения $S_1 \subset S_2$?, так и равенства $S_1 = S_2$? можно выполнить за время

$$T = O(\max\{|S_1|, |S_2|\}) \quad (|S_1|, |S_2| \rightarrow \infty);$$

3) построение множества $\overline{S_1}$ осуществимо за время

$$T = O(|S|) \quad (|S| \rightarrow \infty).$$

Ниже показано, что именно это обстоятельство является основой для построения достаточно мощной алгебраической системы на множестве графов $\bigcup_{n=1}^{\infty} \bigcup_{m=0}^{\binom{n}{2}} \mathbf{G}(n, m)$.

Рассмотрим алгебраическую систему

$$\mathbf{A}_n = (\mathbf{S}_n, \mathbf{F}_{op}^{\mathbf{A}_n}, \mathbf{F}_{rel}^{\mathbf{A}_n}) \quad (n \in \mathbf{N})$$

с носителем

$$\mathbf{S}_n = \bigcup_{m=0}^{\binom{n}{2}} \mathbf{G}(n, m),$$

сигнатурой операций

$$\mathbf{F}_{op}^{\mathbf{A}_n} = \{ \setminus, \cup, \cap, \oplus, \overline{} \}$$

и сигнатурой отношений

$$\mathbf{F}_{rel}^{\mathbf{A}_n} = \{ =, \subset \}.$$

Операции и отношения алгебраической системы \mathbf{A}_n определим следующим образом: если $G_1, G_2 \in \mathbf{S}_n$, где $G_1 = (V, E_1)$ и $G_2 = (V, E_2)$ ($V = \{1, \dots, n\}$), то

$$G_1 \setminus G_2 \stackrel{оп}{=} \{V, E_1 \setminus E_2\},$$

$$G_1 \cup G_2 \stackrel{оп}{=} \{V, E_1 \cup E_2\},$$

$$G_1 \cap G_2 \stackrel{оп}{=} \{V, E_1 \cap E_2\},$$

$$G_1 \oplus G_2 \stackrel{оп}{=} \{V, (E_1 \setminus E_2) \cup (E_2 \setminus E_1)\},$$

$$\overline{G_1} \stackrel{оп}{=} (V, V^{(2)} \setminus E_1),$$

$$G_1 = G_2 \stackrel{оп}{\Leftrightarrow} E_1 = E_2,$$

$$G_1 \subset G_2 \stackrel{оп}{\Leftrightarrow} E_1 \subset E_2.$$

Следует отметить, что именно алгебраическая система \mathbf{A}_n представляет собой основу (используемую в явном или неявном виде), на базе которой осуществляется моделирование процессов, связанных с исследованием *эффективности* и *живучести* сетей (компьютерных, информационных, энергетических и т.д.), имеющих фиксированное множество вершин.

Для любого представления \mathbf{R} графов, принадлежащих множеству \mathbf{S}_n , выполнение операций, принадлежащих множеству $\mathbf{F}_{op}^{\mathbf{A}_n}$, а также про-

верка отношений, принадлежащих множеству $\mathbf{F}_{rel}^{A_n}$, являются задачами из класса $LCl(\mathbf{R})$. Однако, при переходе от одного представления графа к другому может изменяться как время, затрачиваемое на выполнение операций, так и время, затрачиваемое на проверку отношений.

Замечание 2.3. Пусть графы, принадлежащие множеству \mathbf{S}_n , представлены списками смежности ребер (либо матрицами смежности ребер). При выполнении операций, принадлежащих множеству $\mathbf{F}_{op}^{A_n}$, а также проверке отношений, принадлежащих множеству $\mathbf{F}_{rel}^{A_n}$, существенно используется таблица 1.1 каждого из исходных графов.

В случае выполнения операций, именно на основе этих таблиц осуществляется нумерация ребер результирующего графа. Подчеркнем, что, выбор различных нумераций приводит к различным разметкам ребер результирующего графа. Это означает, что конкретный вид списков смежности ребер, представляющих результирующий граф, зависит от способа нумерации ребер результирующего графа. Однако любые два таких представления изоморфны. При этом изоморфизм устанавливается с помощью соответствующих таблиц 1.1. Как было отмечено в п.2.1, построение таблицы 1.1 и работа с ней, не влияет на принадлежность классу $LCl(\mathbf{R})$ как задач выполнения операций, принадлежащих множеству $\mathbf{F}_{op}^{A_n}$, так и задач проверки отношений, принадлежащих множеству $\mathbf{F}_{rel}^{A_n}$.

Пусть \mathbf{R} – стандартная форма представления графов, принадлежащих множеству \mathbf{S}_n , списками смежности вершин. Тогда:

1) для любых графов $G_1, G_2 \in \mathbf{S}_n$ ($G_1 = (V, E_1), G_2 = (V, E_2)$) выполнение как любой из операций $\setminus, \cup, \cap, \oplus$, так и проверка истинности любого из отношений \subset и $=$, осуществимы за время

$$T_n^R(G_1, G_2) = O(\max\{n, \max\{|E_1|, |E_2|\}\}) \quad (n, |E_1|, |E_2| \rightarrow \infty); \quad (2.3)$$

2) для любого графа $G \in \mathbf{S}_n$ построение графа \bar{G} осуществимо за время

$$T_n^R(G) = O(n^2) \quad (n \rightarrow \infty). \quad (2.4)$$

Пусть теперь \mathbf{R} – стандартная форма представления графов, принадлежащих множеству \mathbf{S}_n , списками смежности ребер. Тогда:

1) для любых графов $G_1, G_2 \in \mathbf{S}_n$ ($G_1 = (V, E_1), G_2 = (V, E_2)$) выполнение как любой из операций $\setminus, \cup, \cap, \oplus$, так и проверка истинности любого из отношений \subset и $=$, осуществимы за время

$$T_n^R(G_1, G_2) = O(\max\{|E_1|, |E_2|\} \cdot \max\{1, \min\{n, \max\{|E_1|, |E_2|\}\}\}) \quad ; \quad (2.5)$$

$$(n, |E_1|, |E_2| \rightarrow \infty)$$

2) для любого графа $G \in \mathbf{S}_n$ построение графа \overline{G} осуществимо за время

$$T_n^R(G) = O(n^4) \quad (n \rightarrow \infty). \quad (2.6)$$

Отметим ряд следствий из соотношений (2.3)-(2.6).

Пусть \mathbf{R} – стандартная форма представления графа $G \in \mathbf{S}_n$ списками смежности вершин, либо списками смежности ребер. Из (2.4) и (2.6) вытекает, что время, затрачиваемое на построение стандартной формы \mathbf{R} представления графа \overline{G} , зависит только от числа n вершин графа G и совершенно не зависит от того, сколько и каких именно ребер содержит граф G .

Более существенным является то, что это время, асимптотически равно времени, затрачиваемому на построения соответствующей матрицы смежности графа \overline{G} .

Иная ситуация имеет место при выполнении как любой из операций $\setminus, \cup, \cap, \oplus$, так и проверке истинности любого из отношений $\subset, =$. Пусть \mathbf{R} – стандартная форма представления графов списками смежности вершин. Из (2.3) вытекает, что для представления \mathbf{R} время, затрачиваемое на выполнение как любой из операций $\setminus, \cup, \cap, \oplus$, так и на проверку истинности любого из отношений \subset и $=$, асимптотически меньше, чем соответствующее время для представлений графов матрицами смежности вершин, если

$$\max\{|E_1|, |E_2|\} = o(n^2) \quad (n, |E_1|, |E_2| \rightarrow \infty) \quad (2.7)$$

и это время асимптотически равно соответствующему времени для представлений графов матрицами смежности вершин, если

$$\max\{|E_1|, |E_2|\} = \Theta(n^2) \quad (n, |E_1|, |E_2| \rightarrow \infty). \quad (2.8)$$

Рассмотрим теперь стандартную форму \mathbf{R} представления графов списками смежности ребер. Из (2.5) вытекает, что для представления \mathbf{R} время, затрачиваемое на выполнение как любой из операций $\setminus, \cup, \cap, \oplus$, так и на проверку истинности любого из отношений \subset и $=$, асимптотически меньше, чем соответствующее время для представлений графов матрицами смежности ребер, если

$$n = o(\max\{|E_1|, |E_2|\}) \quad (n, |E_1|, |E_2| \rightarrow \infty). \quad (2.9)$$

Однако не сложно показать, что если либо

$$\max\{|E_1|, |E_2|\} = o(n) \quad (n, |E_1|, |E_2| \rightarrow \infty), \quad (2.10)$$

либо

$$n = \Theta(\max\{|E_1|, |E_2|\}) \quad (n, |E_1|, |E_2| \rightarrow \infty), \quad (2.11)$$

то ситуация может изменяться в зависимости от структуры исходных графов $G_1, G_2 \in \mathbf{S}_n$, над которыми выполняются указанные выше действия.

Для алгебраической системы \mathbf{A}_n значение формул (2.7)-(2.11), устанавливающих соотношения между числом вершин и ребер графов, принадлежащих множеству \mathbf{S}_n , состоит в следующем. Эти соотношения показывают, когда при действиях в рамках алгебраической системы \mathbf{A}_n (т.е. при выполнении операций, принадлежащих множеству $\mathbf{F}_{op}^{\mathbf{A}_n}$, а также при проверке отношений, принадлежащих множеству $\mathbf{F}_{rel}^{\mathbf{A}_n}$) представления стандартными формами списков смежности либо предпочтительнее, чем представления соответствующими матрицами смежности, либо когда эти представления эквивалентны, либо когда для выяснения ситуации требуется осуществить дополнительный анализ. Получим более тонкие характеристики.

Вначале оценим среднее число ребер графа, случайно выбранного из множества \mathbf{S}_n .

Теорема 2.2. Среднее число ребер графа, случайно выбранного из множества \mathbf{S}_n , равно $0.5 \cdot \binom{n}{2}$.

Доказательство. Рассмотрим случайную величину ξ_n , принимающую значение m ($m \in \{0, 1, \dots, \binom{n}{2}\}$) с вероятностью

$$p_{n,m} = \frac{\binom{\binom{n}{2}}{m}}{2^{\binom{n}{2}}},$$

т.е. $p_{n,m}$ ($m \in \{0, 1, \dots, \binom{n}{2}\}$) есть вероятность того, что случайно выбранный из множества \mathbf{S}_n граф содержит m ребер. Найдем математическое ожидание случайной величины ξ_n

$$\mathbf{M}\xi_n = \sum_{m=0}^{\binom{n}{2}} m \cdot p_{n,m} = \frac{1}{2^{\binom{n}{2}}} \cdot \sum_{m=0}^{\binom{n}{2}} m \cdot \binom{\binom{n}{2}}{m} = \frac{1}{2^{\binom{n}{2}}} \cdot \sum_{m=1}^{\binom{n}{2}} m \cdot \binom{\binom{n}{2}}{m} \quad (2.12)$$

Продифференцируем тождество

$$(1+x)^l = \sum_{m=0}^l \binom{l}{m} \cdot x^m.$$

Получим

$$l \cdot (1+x)^{l-1} = \sum_{m=1}^l \binom{l}{m} \cdot m \cdot x^{m-1}. \quad (2.13)$$

Положим в (2.13) $l = \binom{n}{2}$ и $x = 1$. Получим

$$\sum_{m=1}^{\binom{n}{2}} m \cdot \binom{\binom{n}{2}}{m} = \binom{n}{2} \cdot 2^{\binom{n}{2}-1}. \quad (2.14)$$

Подставим (2.14) в (2.12). Получим

$$\mathbf{M}\xi_n = 0.5 \cdot \binom{n}{2}, \quad (2.15)$$

что и требовалось доказать.

Теорема доказана.

Следствие 2.1. Пусть выбор графов, принадлежащих множеству \mathbf{S}_n , осуществляется случайным образом. Тогда, в среднем, время, затрачиваемое на выполнение любой из операций \setminus , \cup , \cap , \oplus , а также на проверку истинности любого из отношений \subset и $=$ над представлениями графов списками смежности вершин, асимптотически равно соответствующему времени для представлений графов матрицами смежности вершин.

Доказательство. Справедливость следствия вытекает из соотношений (2.8), (2.15), а также из равенства

$$\mathbf{M}(\zeta + \eta) = \mathbf{M}(\zeta) + \mathbf{M}(\eta) \quad (2.16)$$

для любых случайных величин ζ и η .

Следствие доказано.

Следствие 2.2. Пусть выбор графов, принадлежащих множеству \mathbf{S}_n , осуществляется случайным образом. Тогда, в среднем, время, затрачиваемое на выполнение любой из операций \setminus , \cup , \cap , \oplus , а также на проверку истинности любого из отношений \subset и $=$ над представлениями графов списками смежности ребер, асимптотически меньше, чем соответствующее время для представлений графов матрицами смежности ребер.

Доказательство. Справедливость следствия вытекает из соотношений (2.9), (2.15) и (2.16).

Следствие доказано.

Замечание 2.4. 1. Представление графов списками смежности вершин применяется (как в теоретических исследованиях, так и при решении прикладных задач) значительно чаще, чем представление графов списками смежности ребер. По-видимому, это вызвано скорее традицией и большей «наглядностью» представления графов списками смежности вершин (как было показано в п.2.1, при представлении графов списками смежности ребер необходимо, также, хранить и информацию вида таблицы 1.1), чем анализом исследуемого множества графов, а также тех действий, которые выполняются над графами. Что же касается сравнения сложности представления графов списками смежности вершин с представлениями списками смежности ребер, а также представлений матрицами смежности вершин с представлениями матрицами смежности ребер, то можно отметить только следующее обстоятельство. Представление графа G , как списками, так и матрицей смежности ребер, по своей сути, означает переход к реберному графу $L(G)$. Следовательно, соотношения между сложностями вершинных и реберных представлений принадлежащих заданному множеству графов, в каждом случае можно получить, только изучив соотношения между соответствиями параметров исходных графов и соответствующих им реберных графов.

2. Следствие 2.1 демонстрирует ситуацию, при которой, в среднем, для представления графов списками смежности вершин нет выигрыша ни по емкостной, ни по временной сложности, по сравнению с представлениями графов матрицами смежности вершин. Сказанное, в частности, характеризует значение алгебры матриц, развитой в примере 1.10. Аналогичным образом, следствие 2.2 демонстрирует ситуацию, при которой, в среднем, представление графов списками смежности ребер дает выигрыш, как по емкостной, так и по временной, сложности, по сравнению с представлением графов матрицами смежности ребер.

Оценим теперь число ребер почти всех графов, принадлежащих множеству \mathbf{S}_n .

Теорема 2.3. Число m ребер почти всех графов, принадлежащих множеству \mathbf{S}_n , удовлетворяет асимптотическому равенству

$$m = \Theta(n^2) \quad (n \rightarrow \infty). \quad (2.17)$$

Доказательство. Вычислим дисперсию случайной величины ξ_n , введенной в теореме 2.2. Воспользуемся формулой

$$\mathbf{D}\xi_n = \mathbf{M}\xi_n^2 - (\mathbf{M}\xi_n)^2. \quad (2.18)$$

Вычислим $\mathbf{M}\xi_n^2$

$$\begin{aligned} \mathbf{M}\xi_n^2 &= \sum_{m=0}^{\binom{n}{2}} m^2 \cdot p_{n,m} = \frac{1}{2^{\binom{n}{2}}} \cdot \sum_{m=0}^{\binom{n}{2}} m^2 \cdot \binom{\binom{n}{2}}{m} = \frac{1}{2^{\binom{n}{2}}} \cdot \sum_{m=1}^{\binom{n}{2}} m^2 \cdot \binom{\binom{n}{2}}{m} = \\ &= \frac{1}{2^{\binom{n}{2}}} \cdot \sum_{m=1}^{\binom{n}{2}} (m \cdot (m-1) + m) \cdot \binom{\binom{n}{2}}{m} = \end{aligned}$$

$$\begin{aligned}
&= \frac{1}{2^{\binom{n}{2}}} \cdot \sum_{m=2}^{\binom{n}{2}} m \cdot (m-1) \cdot \binom{\binom{n}{2}}{m} + \frac{1}{2^{\binom{n}{2}}} \cdot \sum_{m=1}^{\binom{n}{2}} m \cdot \binom{\binom{n}{2}}{m} = \\
&= \frac{1}{2^{\binom{n}{2}}} \cdot \sum_{m=2}^{\binom{n}{2}} m \cdot (m-1) \cdot \binom{\binom{n}{2}}{m} + \mathbf{M}\xi_n.
\end{aligned} \tag{2.19}$$

Продифференцируем тождество (2.13). Получим

$$l \cdot (l-1) \cdot (1+x)^{l-2} = \sum_{m=2}^l \binom{l}{m} \cdot m \cdot (m-1) \cdot x^{m-2}. \tag{2.20}$$

Положим в (2.20) $l = \binom{n}{2}$ и $x = 1$. Получим

$$\sum_{m=2}^{\binom{n}{2}} m \cdot (m-1) \cdot \binom{\binom{n}{2}}{m} = \binom{\binom{n}{2}}{2} \cdot \left(\binom{\binom{n}{2}}{2} - 1 \right) \cdot 2^{\binom{n}{2}-2}. \tag{2.21}$$

Подставим (2.21) и (2.15) в (2.19). Получим

$$\begin{aligned}
\mathbf{M}\xi_n^2 &= 0.25 \cdot \binom{\binom{n}{2}}{2} \cdot \left(\binom{\binom{n}{2}}{2} - 1 \right) + 0.5 \cdot \binom{\binom{n}{2}}{2} = \\
&= 0.25 \cdot \binom{\binom{n}{2}}{2}^2 + 0.25 \cdot \binom{\binom{n}{2}}{2}.
\end{aligned} \tag{2.22}$$

Подставим (2.22) и (2.15) в (2.18). Получим

$$\mathbf{D}\xi_n = 0.25 \cdot \binom{\binom{n}{2}}{2}^2 + 0.25 \cdot \binom{\binom{n}{2}}{2} - 0.25 \cdot \binom{\binom{n}{2}}{2}^2 = 0.25 \cdot \binom{\binom{n}{2}}{2}. \tag{2.23}$$

Применим к случайной величине ξ_n неравенство Чебышева

$$P\{|\xi_n - \mathbf{M}\xi_n| \geq t\} \leq \frac{\mathbf{D}\xi_n}{t^2}. \tag{2.24}$$

Положим

$$\frac{\mathbf{D}\xi_n}{t^2} = \varepsilon. \tag{2.25}$$

По определению случайной величины ξ_n , левая часть неравенства (2.24) равна доле графов, принадлежащих множеству \mathbf{S}_n , для которых число ребер m удовлетворяет неравенству

$$|m - 0.5 \cdot \binom{n}{2}| \geq t. \quad (2.26)$$

Следовательно, доля графов, принадлежащих множеству \mathbf{S}_n , для которых число ребер m удовлетворяет неравенству

$$|m - 0.5 \cdot \binom{n}{2}| < t,$$

или, что эквивалентно, неравенствам

$$0.5 \cdot \binom{n}{2} - t < m < 0.5 \cdot \binom{n}{2} + t, \quad (2.27)$$

не меньше, чем

$$E(\varepsilon) = 1 - \varepsilon. \quad (2.28)$$

Из равенства (2.25), с учетом равенства (2.23), находим

$$t = \frac{0.5}{\sqrt{\varepsilon}} \cdot \sqrt{\binom{n}{2}}. \quad (2.29)$$

Подставим (2.29) в (2.27). Получаем, что доля графов, принадлежащих множеству \mathbf{S}_n , для которых число ребер m удовлетворяет неравенствам

$$0.5 \cdot \binom{n}{2} - \frac{0.5}{\sqrt{\varepsilon}} \cdot \sqrt{\binom{n}{2}} < m < 0.5 \cdot \binom{n}{2} + \frac{0.5}{\sqrt{\varepsilon}} \cdot \sqrt{\binom{n}{2}}, \quad (2.30)$$

не меньше, чем

$$E(\varepsilon) = 1 - \varepsilon. \quad (2.31)$$

Положим

$$\varepsilon = \Theta(n^{-2\alpha}) \quad (n \rightarrow \infty), \quad (2.32)$$

где

$$\alpha \in (0;1). \quad (2.33)$$

Из (2.32) вытекает, что

$$\frac{0.5}{\sqrt{\varepsilon}} \cdot \sqrt{\binom{n}{2}} = \Theta(n^{1+\alpha}) \quad (n \rightarrow \infty), \quad (2.34)$$

а из (2.28), (2.32) и (2.33) – что

$$E(\varepsilon) \rightarrow 1 \quad (n \rightarrow \infty). \quad (2.35)$$

Следовательно, из (2.30), (2.34) и (2.35) вытекает, что число m ребер почти всех графов, принадлежащих множеству \mathbf{S}_n , удовлетворяет асимптотическому равенству $m = \Theta(n^2)$ ($n \rightarrow \infty$), что и требовалось доказать.

Теорема доказана.

Следствие 2.3. Для почти всех графов, принадлежащих множеству \mathbf{S}_n ($n \rightarrow \infty$), время, затрачиваемое на выполнение любой из операций \setminus , \cup , \cap , \oplus , а также на проверку истинности любого из отношений \subset и $=$ над представлениями графов списками смежности вершин, асимптотически равно соответствующему времени для представлений графов матрицами смежности вершин.

Доказательство. Справедливость следствия вытекает из соотношений (2.8) и (2.17).

Следствие доказано.

Следствие 2.4. Для почти всех графов, принадлежащих множеству \mathbf{S}_n ($n \rightarrow \infty$), время, затрачиваемое на выполнение любой из операций \setminus , \cup , \cap , \oplus , а также на проверку истинности любого из отношений \subset и $=$ над представлениями графов списками смежности ребер, асимптотически меньше, чем соответствующее время для представлений графов матрицами смежности ребер.

Доказательство. Справедливость следствия вытекает из соотношений (2.9) и (2.17).

Следствие доказано.

Замечание 2.5. Из следствия 2.3 вытекает, что при действиях в рамках алгебраической системы \mathbf{A}_n на всем множестве \mathbf{S}_n (а не на некоторых его подмножествах) почти во всех случаях представление графов списками смежности вершин не дает выигрыша ни по емкостной, ни по временной сложности, по сравнению с представлением графов матрицей смежности вершин. В тоже время из следствия 2.4 вытекает, что при тех же условиях почти во всех случаях представление графов списками смежности ребер дает выигрыш, как по емкостной, так и по временной, сложности, по сравнению с представлением графов матрицами смежности ребер.

Перейдем от алгебраической системы \mathbf{A}_n к алгебраической системе на множестве графов $\bigcup_{n=1}^{\infty} \bigcup_{m=0}^{\binom{n}{2}} \mathbf{G}(n, m)$, содержащей все основные операции над графами (см., напр., [12]).

Замечание 2.6. 1. При построении алгебраической системы на множестве графов $\bigcup_{n=1}^{\infty} \bigcup_{m=0}^{\binom{n}{2}} \mathbf{G}(n, m)$ воспользуемся стандартной техникой погружения одной алгебраической системы в другую, т.е. \mathbf{A}_n рассматривается как подсистема новой алгебраической системы. Такой подход дает возможность непосредственно использовать все полученные выше результаты, связанные со сложностью выполнения операций, принадлежащих множеству $\mathbf{F}_{op}^{\mathbf{A}_n}$ и проверки отношений, принадлежащих множеству $\mathbf{F}_{rel}^{\mathbf{A}_n}$.

2. Так как речь идет о построении алгебраической системы на множестве графов $\bigcup_{n=1}^{\infty} \bigcup_{m=0}^{\binom{n}{2}} \mathbf{G}(n, m)$, то результат любой операции над графами, принадлежащими множеству $\bigcup_{n=1}^{\infty} \bigcup_{m=0}^{\binom{n}{2}} \mathbf{G}(n, m)$, также должен быть графом, принадлежащим множеству $\bigcup_{n=1}^{\infty} \bigcup_{m=0}^{\binom{n}{2}} \mathbf{G}(n, m)$. Отсюда вытекает, что в определение операций конструируемой алгебраической системы заведомо включается некоторая фиксированная нумерация вершин результирующего графа. Такой подход не ограничивает общности рассуждений, так как при изменении нумерации вершин результирующего графа получается изоморфный ему граф. Последнее означает, что при изменении нумерации вершин результирующего графа сохраняются все абстрактные свойства конструируемой алгебраической системы.

Рассмотрим алгебраическую систему

$$\mathbf{B} = (\mathbf{T}, \mathbf{F}_{op}^{\mathbf{B}}, \mathbf{F}_{rel}^{\mathbf{B}})$$

с носителем

$$\mathbf{T} = \bigcup_{n=1}^{\infty} \bigcup_{m=0}^{\binom{n}{2}} \mathbf{G}(n, m),$$

сигнатурой операций

$$\mathbf{F}_{op}^{\mathbf{B}} = \left(\bigcup_{n=1}^{\infty} \mathbf{F}_{op}^{\mathbf{A}_n} \right) \cup \{ \circ, \rightarrow, \bullet, *, \wedge, \vee \}$$

и сигнатурой отношений

$$\mathbf{F}_{rel}^{\mathbf{B}} = \left(\bigcup_{n=1}^{\infty} \mathbf{F}_{rel}^{\mathbf{A}_n} \right) \cup \{ \leq, \triangleleft \}.$$

Операции и отношения алгебраической системы \mathbf{B} определим следующим образом. Пусть $G_1 = (V_1, E_1) \in \mathbf{T}$ и $G_2 = (V_2, E_2) \in \mathbf{T}$. Тогда:

$$1) G_1 \circ_{op} G_2 = G, \text{ где}$$

$$G = (V, E) \in \mathbf{G}(|V_1| + |V_2|, |E_1| + |E_2|), \quad (2.36)$$

причем

$$E = E_1 \cup \{\{i+|V_1|, j+|V_1|\} \mid \{i, j\} \in E_2\}; \quad (2.37)$$

2) $G_1 \rightarrow G_2 = G$, где

$$G = (V, E) \in \mathbf{G}(|V_1| + |V_2|, |E_1| + |E_2| + |V_1| \cdot |V_2|), \quad (2.38)$$

причем

$$E = E_1 \cup \{\{i+|V_1|, j+|V_1|\} \mid \{i, j\} \in E_2\} \cup \{\{i, j+|V_1|\} \mid i \in V_1, j \in V_2\}; \quad (2.39)$$

3) $G_1 \bullet G_2 = G$, где

$$G = (V, E) \in \mathbf{G}(|V_1| \cdot |V_2|, |V_1| \cdot |E_2| + |V_2| \cdot |E_1|), \quad (2.40)$$

причем, если $|V_1| \leq |V_2|$, то

$$\begin{aligned} & \{\alpha + \beta \cdot |V_2|, \delta + \gamma \cdot |V_2|\} \in E \Leftrightarrow \\ & \Leftrightarrow ((\alpha = \delta) \& (\{\beta + 1, \gamma + 1\} \in E_1)) \vee ((\beta = \gamma) \& (\{\alpha, \delta\} \in E_2)) \end{aligned} \quad (2.41)$$

для всех $\alpha, \delta \in \{1, \dots, |V_2|\}$ и $\beta, \gamma \in \{0, 1, \dots, |V_1| - 1\}$, а если $|V_1| > |V_2|$, то

$$\begin{aligned} & \{\alpha + \beta \cdot |V_1|, \delta + \gamma \cdot |V_1|\} \in E \Leftrightarrow \\ & \Leftrightarrow ((\alpha = \delta) \& (\{\beta + 1, \gamma + 1\} \in E_2)) \vee ((\beta = \gamma) \& (\{\alpha, \delta\} \in E_1)) \end{aligned} \quad (2.41')$$

для всех $\alpha, \delta \in \{1, \dots, |V_1|\}$ и $\beta, \gamma \in \{0, 1, \dots, |V_2| - 1\}$;

4) $G_1 * G_2 = G$, где

$$G = (V, E) \in \mathbf{G}(|V_1| \cdot |V_2|, |V_1| \cdot |E_2| + |V_2|^2 \cdot |E_1|), \quad (2.42)$$

причем, если $|V_1| \leq |V_2|$, то

$$\begin{aligned} & \{\alpha + \beta \cdot |V_2|, \delta + \gamma \cdot |V_2|\} \in E \Leftrightarrow \\ & \Leftrightarrow (\{\beta + 1, \gamma + 1\} \in E_1) \vee ((\beta = \gamma) \& (\{\alpha, \delta\} \in E_2)) \end{aligned} \quad (2.43)$$

для всех $\alpha, \delta \in \{1, \dots, |V_2|\}$ и $\beta, \gamma \in \{0, 1, \dots, |V_1| - 1\}$, а если $|V_1| > |V_2|$, то

$$\begin{aligned} & \{\alpha + \beta \cdot |V_1|, \delta + \gamma \cdot |V_1|\} \in E \Leftrightarrow \\ & \Leftrightarrow (\{\alpha, \delta\} \in E_1) \vee ((\alpha = \delta) \& (\{\beta + 1, \gamma + 1\} \in E_2)) \end{aligned} \quad (2.43')$$

для всех $\alpha, \delta \in \{1, \dots, |V_1|\}$ и $\beta, \gamma \in \{0, 1, \dots, |V_2| - 1\}$;

5) $G_1 \wedge G_2 = G$, где

$$G = (V, E) \in \mathbf{G}(|V_1| \cdot |V_2|, |E_1| \cdot |E_2|), \quad (2.44)$$

причем, если $|V_1| \leq |V_2|$, то

$$\{\alpha + \beta \cdot |V_2|, \delta + \gamma \cdot |V_2|\} \in E \Leftrightarrow (\{\beta + 1, \gamma + 1\} \in E_1) \& (\{\alpha, \delta\} \in E_2) \quad (2.45)$$

для всех $\alpha, \delta \in \{1, \dots, |V_2|\}$ и $\beta, \gamma \in \{0, 1, \dots, |V_1| - 1\}$, а если $|V_1| > |V_2|$, то

$$\{\alpha + \beta \cdot |V_1|, \delta + \gamma \cdot |V_1|\} \in E \Leftrightarrow (\{\alpha, \delta\} \in E_1) \& (\{\beta + 1, \gamma + 1\} \in E_2) \quad (2.45')$$

для всех $\alpha, \delta \in \{1, \dots, |V_1|\}$ и $\beta, \gamma \in \{0, 1, \dots, |V_2| - 1\}$;

6) $G_1 \vee G_2 = G$, где

$$G = (V, E) \in \mathbf{G}(|V_1| \cdot |V_2|, |E_1| \cdot |V_2|^2 + |E_2| \cdot |V_1|^2 - |E_1| \cdot |E_2|), \quad (2.46)$$

причем, если $|V_1| \leq |V_2|$, то

$$\{\alpha + \beta \cdot |V_2|, \delta + \gamma \cdot |V_2|\} \in E \Leftrightarrow (\{\beta + 1, \gamma + 1\} \in E_1 \vee (\{\alpha, \delta\} \in E_2)) \quad (2.47)$$

для всех $\alpha, \delta \in \{1, \dots, |V_2|\}$ и $\beta, \gamma \in \{0, 1, \dots, |V_1| - 1\}$, а если $|V_1| > |V_2|$, то

$$\{\alpha + \beta \cdot |V_1|, \delta + \gamma \cdot |V_1|\} \in E \Leftrightarrow (\{\alpha, \delta\} \in E_1) \vee (\{\beta + 1, \gamma + 1\} \in E_2) \quad (2.47')$$

для всех $\alpha, \delta \in \{1, \dots, |V_1|\}$ и $\beta, \gamma \in \{0, 1, \dots, |V_2| - 1\}$;

7) $G_1 \leq G_2$ тогда и только тогда, когда G_1 – подграф графа G_2 ;

8) $G_1 \triangleleft G_2$ тогда и только тогда, когда G_2 содержит подграф, изоморфный графу G_1 .

Замечание 2.7. 1. Определенные выше операции \circ , \rightarrow , \bullet , $*$, \wedge представляют собой модификации операций *объединения*, *соединения*, *произведения*, *композиции* и *конъюнкции*, определяемых, как правило, для графов с непересекающимися множествами вершин (см., напр., [12]).

2. Операция \vee , определенная соотношениями (2.46), (2.47) и (2.47'), как и операция \wedge , определенная соотношениями (2.45), (2.46) и (2.46'), имеет многочисленные применения, связанные с построением и анализом алгоритмов, в частности, связанные с распараллеливанием вычислений. Действительно, пусть исходные графы G_1 и G_2 соответствуют решаемым *подзадачам*, их вершины соответствуют отдельным *этапам* решения (или, иными словами, соответствуют *состояниям*), а ребра указывают, какие из этих ситуаций – *соседние*. Тогда граф $G_1 \wedge G_2$ описывает ситуацию, когда допустимо только одновременное продвижение по этапам обеих решаемых подзадач, а граф $G_1 \vee G_2$ – ситуацию, когда допустимо продвижение по этапам хотя бы одной из решаемых подзадач.

3. Нетрудно убедиться в том, что операции алгебраической системы **B** удовлетворяют следующим условиям:

1) операции \circ и \rightarrow ассоциативны, т.е.

$$G_1 \circ (G_2 \circ G_3) = (G_1 \circ G_2) \circ G_3, \quad (2.48)$$

$$G_1 \rightarrow (G_2 \rightarrow G_3) = (G_1 \rightarrow G_2) \rightarrow G_3 \quad (2.49)$$

для всех $G_1, G_2, G_3 \in \mathbf{T}$;

2) ни одна из операций \bullet , $*$, \wedge , \vee не ассоциативна, однако,

$$G_1 \bullet (G_2 \bullet G_3) \sim (G_1 \bullet G_2) \bullet G_3, \quad (2.50)$$

$$G_1 * (G_2 * G_3) \sim (G_1 * G_2) * G_3, \quad (2.51)$$

$$G_1 \wedge (G_2 \wedge G_3) \sim (G_1 \wedge G_2) \wedge G_3, \quad (2.52)$$

$$G_1 \vee (G_2 \vee G_3) \sim (G_1 \vee G_2) \vee G_3 \quad (2.53)$$

для всех $G_1, G_2, G_3 \in \mathbf{T}$, где \sim – отношение изоморфизма графов, принадлежащих множеству \mathbf{T} ;

3) ни одна из операций \circ , \rightarrow , \bullet , $*$, \wedge , \vee не коммутативна, однако,

$$G_1 \circ G_2 \sim G_2 \circ G_1, \quad (2.54)$$

$$G_1 \rightarrow G_2 \sim G_2 \rightarrow G_1, \quad (2.55)$$

$$G_1 \bullet G_2 \sim G_2 \bullet G_1, \quad (2.56)$$

$$G_1 \wedge G_2 \sim G_2 \wedge G_1, \quad (2.57)$$

$$G_1 \vee G_2 \sim G_2 \vee G_1 \quad (2.58)$$

для всех $G_1, G_2, G_3 \in \mathbf{T}$.

4. Определим на фактор-множестве \mathbf{T} / \sim операции \circ , \rightarrow , \bullet , $*$, \wedge , \vee и отношение \triangleleft в соответствии с равенствами

$$[G_1] \diamond [G_2] = [G_1 \diamond G_2] \quad (\diamond \in \{\circ, \rightarrow, \bullet, *, \wedge, \vee\}),$$

$$[G_1] \triangleleft [G_2] \Leftrightarrow (\forall G' \in [G_1])(\forall G'' \in [G_2])(G' \triangleleft G''),$$

где $[G]$ – класс отношения изоморфизма \sim , содержащий граф G . Таким образом, мы приходим к алгебраической системе

$$\mathbf{C} = (\mathbf{T} / \sim, \{\circ, \rightarrow, \bullet, *, \wedge, \vee\}, \{\triangleleft, =\}),$$

в которой все операции \circ , \rightarrow , \bullet , $*$, \wedge , \vee – ассоциативные, а операции \circ , \rightarrow , \bullet , \wedge , \vee , также, и коммутативные, что непосредственно вытекает из (2.48)-

(2.58). Отметим, что отношение \triangleleft играет в алгебраической системе \mathbf{C} роль отношения *частичного порядка*.

Исследуем сложность выполнения операций алгебраической системы \mathbf{B} . Пусть $G_1 = (V_1, E_1) \in \mathbf{T}$ и $G_2 = (V_2, E_2) \in \mathbf{T}$. Из (2.36)-(2.39) вытекает, что задачи выполнения, как операции \circ , так и операции \rightarrow , принадлежат классу $LCl(\mathbf{R})$ для любого представления \mathbf{R} графов, принадлежащих множеству \mathbf{T} . Более того, из (2.36)-(2.39) вытекает, что операция \circ выполнима за время

$$T_{\circ} = O(|V_1| + |V_2| + |E_1| + |E_2|) \quad (|V_1|, |V_2|, |E_1|, |E_2| \rightarrow \infty),$$

т.е. за время, линейное от размера входа, а операция \rightarrow выполнима за время

$$T_{\rightarrow} = O((|V_1| + |V_2|)^2) \quad (|V_1|, |V_2| \rightarrow \infty).$$

Из (2.40), (2.42), (2.44) и (2.46) вытекает, что ни одна из задач выполнения операций \bullet , $*$, \wedge , \vee не принадлежит классу $LCl(\mathbf{R})$ ни при одном представлении \mathbf{R} графов, принадлежащих множеству \mathbf{T} (чтобы убедиться в этом, достаточно построить граф $K_n \diamond K_n$ ($\diamond \in \{\bullet, *, \wedge, \vee\}$)). Однако, справедлива следующая характеристика емкостной сложности этих операций.

Теорема 2.4. Задачи выполнения операций \bullet , $*$, \wedge , \vee алгебраической системы \mathbf{B} принадлежат классу $WORK - LCl(\mathbf{R})$, где \mathbf{R} – представление графов, принадлежащих множеству \mathbf{T} , либо списками, либо матрицами смежности (либо вершин, либо ребер).

Доказательство. Пусть $G_1 = (V_1, E_1) \in \mathbf{T}_n$ и $G_2 = (V_2, E_2) \in \mathbf{T}_n$, где

$$\mathbf{T}_n = \bigcup_{k=1}^n \bigcup_{m=0}^k \mathbf{G}(k, m). \quad (2.59)$$

Пусть \mathbf{R} – представление графов, принадлежащих множеству \mathbf{T} либо списками, либо матрицами смежности вершин. Тогда для каждой из операций \bullet , $*$, \wedge , \vee :

1) задача нумерации вершин графа $G_1 \diamond G_2$ ($\diamond \in \{\bullet, *, \wedge, \vee\}$) принадлежат классу $LCl(\mathbf{R})$ (для каждой операции соответствующий алгоритм, осуществляющий нумерацию вершин результирующего графа $G_1 \diamond G_2$, строится на основе соотношений (2.40), (2.42), (2.44) и (2.46));

2) задача проверки смежности любых двух вершин графа $G_1 \diamond G_2$ ($\diamond \in \{\bullet, *, \wedge, \vee\}$) принадлежит классу $LCl(\mathbf{R})$ (для каждой операции соответствующий алгоритм, осуществляющий проверку смежности вер-

шин результирующего графа $G_1 \diamond G_2$, строится на основе соотношений (2.41), (2.41'), (2.43), (2.43'), (2.45), (2.45') и (2.47), (2.47')).

Следовательно, классу $WORK - LCl(\mathbf{R})$ принадлежат как задача последовательной генерации списков смежности вершин графа $G_1 \diamond G_2$ ($\diamond \in \{ \bullet, *, \wedge, \vee \}$), так и задача последовательной, построчной, генерации матрицы смежности вершин графа $G_1 \diamond G_2$ ($\diamond \in \{ \bullet, *, \wedge, \vee \}$), что и требовалось показать.

Пусть \mathbf{R} – представление графов, принадлежащих множеству \mathbf{T}_n , либо списками, либо матрицами смежности ребер. Для каждого из графов G_1 и G_2 задана таблица 1.1, работа с которой не выводит за пределы класса $LCl(\mathbf{R})$. Следовательно, как задача нумерации ребер графа $G_1 \diamond G_2$ ($\diamond \in \{ \bullet, *, \wedge, \vee \}$), так и задача проверки смежности любых двух ребер графа $G_1 \diamond G_2$ ($\diamond \in \{ \bullet, *, \wedge, \vee \}$) принадлежат классу $LCl(\mathbf{R})$ (обе эти задачи решаются на основе соотношений (2.40)-(2.47), (2.41'), (2.43'), (2.45'), и (2.47') и таблиц 1.1). Отсюда вытекает, что классу $WORK - LCl(\mathbf{R})$ принадлежат как задача последовательной генерации списков смежности ребер графа $G_1 \diamond G_2$ ($\diamond \in \{ \bullet, *, \wedge, \vee \}$), так и задача построчной генерации матрицы смежности ребер $G_1 \diamond G_2$ ($\diamond \in \{ \bullet, *, \wedge, \vee \}$), что и требовалось показать.

Теорема доказана.

Теорема 2.5. Среднее число вершин n_{cp} графа, случайно выбранного из множества \mathbf{T}_n , удовлетворяет асимптотическому равенству

$$n_{cp} = \Theta(n) \quad (n \rightarrow \infty).$$

Доказательство. Рассмотрим случайную величину η_n , принимающую значение k ($k \in \{1, \dots, n\}$) с вероятностью

$$p_{n,k} = \frac{2^{\binom{k}{2}}}{|\mathbf{T}_n|},$$

т.е. $p_{n,k}$ ($k \in \{1, \dots, n\}$) есть вероятность того, что случайно выбранный из множества \mathbf{T}_n граф содержит k вершин. Найдем математическое ожидание случайной величины η_n

$$\mathbf{M}\eta_n = \sum_{k=1}^n k \cdot p_{n,k} = \sum_{k=1}^n k \cdot \frac{2^{\binom{k}{2}}}{|\mathbf{T}_n|} = \frac{1}{|\mathbf{T}_n|} \cdot \sum_{k=1}^n k \cdot 2^{\binom{k}{2}} \quad (2.60)$$

Из (2.59) вытекает, что

$$|\mathbf{T}_n| = \sum_{k=1}^n 2^{\binom{k}{2}} = 2^{\binom{n}{2}} \cdot \sum_{k=1}^n 2^{\binom{k}{2} - \binom{n}{2}} = 2^{\binom{n}{2}} \cdot \sum_{k=1}^n \left(2^{-\frac{n+k-1}{2}} \right)^{n-k} \quad (2.61)$$

Так как $k \in \{1, \dots, n\}$, то

$$0.5 \cdot n < \frac{n+k-1}{2} < n-0.5. \quad (2.62)$$

Из (2.61) и (2.62) вытекает, что

$$2^{\binom{n}{2}} \cdot \sum_{k=1}^n b^{n-k} < |\mathbf{T}_n| < 2^{\binom{n}{2}} \cdot \sum_{k=1}^n a^{n-k}, \quad (2.63)$$

где

$$a = (\sqrt{2})^{-n} \quad (2.64)$$

и

$$b = \sqrt{2} \cdot 2^{-n}. \quad (2.65)$$

Неравенства (2.63) могут быть представлены в следующем, более компактном виде

$$2^{\binom{n}{2}} \cdot \frac{1-b^n}{1-b} < |\mathbf{T}_n| < 2^{\binom{n}{2}} \cdot \frac{1-a^n}{1-a}. \quad (2.66)$$

Далее, имеем очевидное неравенство

$$\sum_{k=1}^n k \cdot 2^{\binom{k}{2}} \geq n \cdot 2^{\binom{n}{2}}. \quad (2.67)$$

Из (2.60), (2.66) и (2.67) получаем

$$\mathbf{M}\eta_n > \frac{n \cdot 2^{\binom{n}{2}}}{2^{\binom{n}{2}} \cdot \frac{1-a^n}{1-a}} = n \cdot \frac{1-a}{1-a^n}. \quad (2.68)$$

Так как множеством значений случайной величины η_n является множество $\{1, \dots, n\}$, то

$$\mathbf{M}\eta_n \leq n. \quad (2.69)$$

Из (2.68) и (2.69) вытекает, что

$$n \cdot \frac{1-a}{1-a^n} < \mathbf{M}\eta_n \leq n, \quad (2.70)$$

где $a \rightarrow 0$ ($n \rightarrow \infty$) (что вытекает из (2.64)). Из (2.70) непосредственно вытекает, что среднее число вершин графа, случайно выбранного из множества \mathbf{T}_n , удовлетворяет асимптотическому равенству $n_{cp} = \Theta(n)$ ($n \rightarrow \infty$), что и требовалось доказать.

Теорема доказана.

Следствие 2.5. Среднее число ребер m_{cp} графа, случайно выбранного из множества \mathbf{T}_n , удовлетворяет асимптотическому равенству

$$m_{cp} = \Theta(n^2) \quad (n \rightarrow \infty).$$

Доказательство. Из теоремы 2.5 вытекает, что среднее число вершин случайно выбранного из множества \mathbf{T}_n графа равно $\Theta(n)$ ($n \rightarrow \infty$). Это эквивалентно тому, что случайный выбор графа осуществляется из множества $\bigcup_{k=\lfloor c_1 \cdot n \rfloor}^{\lceil c_2 \cdot n \rceil} \mathbf{S}_k$ ($n \rightarrow \infty$), где c_1, c_2 – некоторые константы, удовлетворяющие неравенствам $0 < c_1 < c_2 \leq 1$.

В силу теоремы 2.2, если $m_{cp}(\mathbf{S}_k)$ – среднее число ребер графа, случайно выбранного из множества \mathbf{S}_k , то

$$m_{cp}(\mathbf{S}_k) = 0.5 \cdot \binom{k}{2}. \quad (2.71)$$

Пусть $m_{cp}(\bigcup_{k=\lfloor c_1 \cdot n \rfloor}^{\lceil c_2 \cdot n \rceil} \mathbf{S}_k)$ ($n \rightarrow \infty$) – среднее число ребер графа, случайно выбранного из множества $\bigcup_{k=\lfloor c_1 \cdot n \rfloor}^{\lceil c_2 \cdot n \rceil} \mathbf{S}_k$ ($n \rightarrow \infty$). Так как множества \mathbf{S}_k ($k \in \{\lfloor c_1 \cdot n \rfloor, \dots, \lceil c_2 \cdot n \rceil\}$) попарно не пересекаются, то

$$m_{cp}(\bigcup_{k=\lfloor c_1 \cdot n \rfloor}^{\lceil c_2 \cdot n \rceil} \mathbf{S}_k) = \sum_{k=\lfloor c_1 \cdot n \rfloor}^{\lceil c_2 \cdot n \rceil} \alpha_k \cdot m_{cp}(\mathbf{S}_k) \quad (n \rightarrow \infty), \quad (2.72)$$

где

$$(\forall k \in \{\lfloor c_1 \cdot n \rfloor, \dots, \lceil c_2 \cdot n \rceil\})(\alpha_k > 0) \ \& \ (\sum_{k=\lfloor c_1 \cdot n \rfloor}^{\lceil c_2 \cdot n \rceil} \alpha_k = 1) \quad (n \rightarrow \infty). \quad (2.73)$$

Из (2.71)-(2.73) вытекает, что

$$m_{cp}(\bigcup_{k=\lfloor c_1 \cdot n \rfloor}^{\lceil c_2 \cdot n \rceil} \mathbf{S}_k) = \Omega(n^2) \quad (n \rightarrow \infty), \quad (2.74)$$

Число ребер любого графа $G \in \mathbf{T}_n$, не превосходит величины $\binom{n}{2}$.

Следовательно,

$$m_{cp} \left(\bigcup_{k=\lfloor c_1 \cdot n \rfloor}^{\lceil c_2 \cdot n \rceil} \mathbf{S}_k \right) = \Omega(n^2) \quad (n \rightarrow \infty). \quad (2.75)$$

Из (2.74) и (2.75) вытекает, что $m_{cp} \left(\bigcup_{k=\lfloor c_1 \cdot n \rfloor}^{\lceil c_2 \cdot n \rceil} \mathbf{S}_k \right) = \Theta(n^2) \quad (n \rightarrow \infty)$, что и требовалось доказать.

Следствие доказано.

Следствие 2.6. Среднее время T_\diamond ($\diamond \in \{\circ, \rightarrow, \bullet, *, \wedge, \vee\}$) выполнения операции \diamond над графами, случайно выбираемыми из множества \mathbf{T}_n , удовлетворяет следующим асимптотическим равенствам

$$T_\circ = \Omega(n^2) \quad (n \rightarrow \infty) \quad (2.76)$$

$$T_{\rightarrow} = \Omega(n^2) \quad (n \rightarrow \infty) \quad (2.77)$$

$$T_\bullet = \Omega(n^3) \quad (n \rightarrow \infty) \quad (2.78)$$

$$T_* = \Omega(n^4) \quad (n \rightarrow \infty) \quad (2.79)$$

$$T_\wedge = \Omega(n^4) \quad (n \rightarrow \infty) \quad (2.80)$$

$$T_\vee = \Omega(n^4) \quad (n \rightarrow \infty) \quad (2.81)$$

Доказательство. Из теоремы 2.5, следствия 2.5 и формул (2.36), (2.38), (2.40), (2.42), (2.44) и (2.46) вытекает, что при выполнении операции $\diamond \in \{\circ, \rightarrow, \bullet, *, \wedge, \vee\}$ над графами G_1 и G_2 , случайно выбираемыми из множества \mathbf{T}_n , в среднем, имеют место следующие соотношения

$$G_1 \circ G_2 \in \mathbf{G}(\Theta(n), \Theta(n^2)) \quad (n \rightarrow \infty), \quad (2.82)$$

$$G_1 \rightarrow G_2 \in \mathbf{G}(\Theta(n), \Theta(n^2)) \quad (n \rightarrow \infty), \quad (2.83)$$

$$G_1 \bullet G_2 \in \mathbf{G}(\Theta(n^2), \Theta(n^3)) \quad (n \rightarrow \infty), \quad (2.84)$$

$$G_1 * G_2 \in \mathbf{G}(\Theta(n^2), \Theta(n^4)) \quad (n \rightarrow \infty), \quad (2.85)$$

$$G_1 \wedge G_2 \in \mathbf{G}(\Theta(n^2), \Theta(n^4)) \quad (n \rightarrow \infty), \quad (2.86)$$

$$G_1 \vee G_2 \in \mathbf{G}(\Theta(n^2), \Theta(n^4)) \quad (n \rightarrow \infty). \quad (2.87)$$

Асимптотические равенства (2.76)-(2.81) непосредственно вытекают из асимптотических равенств (2.82)-(2.87), что и требовалось доказать.

Следствие доказано.

Исследуем сложность проверки отношений \leq и \triangleleft алгебраической системы **B**.

Теорема 2.6. Задачи проверки отношений \leq , \triangleleft алгебраической системы **B** принадлежат классу $Lcl(\mathbf{R})$, где **R** – представление графов, принадлежащих множеству **T**, либо списками, либо матрицами смежности (либо вершин, либо ребер).

Доказательство. Пусть $G_1 = (V_1, E_1) \in \mathbf{T}_n$ и $G_2 = (V_2, E_2) \in \mathbf{T}_n$. Рассмотрим задачи проверки отношений $G_1 \leq G_2 ?$ и $G_1 \triangleleft G_2 ?$. Без ограничения общности считаем, что $|V_1| \leq |V_2|$ и $|E_1| \leq |E_2|$, так как

$$(|V_1| > |V_2|) \vee (|E_1| > |E_2|) \Rightarrow (G_1 \not\leq G_2) \& (G_1 \not\triangleleft G_2),$$

1. Рассмотрим задачу проверки отношения $G_1 \leq G_2 ?$.

Пусть **R** – представление графов, принадлежащих множеству **T**, списками смежности вершин. Из представления графа G_2

$$LIST_{G_2} : LIST_1^{G_2}, LIST_2^{G_2}, \dots, LIST_{|V_2|}^{G_2}$$

выделим первые $|V_1|$ списков. Удалим из каждого выделенного списка все элементы, не принадлежащие множеству V_1 . Получим последовательность списков

$$LIST : LIST_1, LIST_2, \dots, LIST_{|V_1|},$$

представляющих подграф G графа G_2 , индуцированный множеством вершин V_1 . Теперь для последовательности **LIST** и для представления

$$LIST_{G_1} : LIST_1^{G_1}, LIST_2^{G_1}, \dots, LIST_{|V_1|}^{G_1}$$

графа G_1 достаточно проверить истинность высказывания

$$(\forall i \in \{1, \dots, |V_1|\})(LIST_i^{G_1} \subseteq LIST_i).$$

Ясно, что если это высказывание – истинное, то $G_1 \leq G_2$, иначе $G_1 \not\leq G_2$. Все описанные действия принадлежат классу $Lcl(\mathbf{R})$, где **R** – представление графов, принадлежащих множеству **T**, списками смежности вершин. Следовательно, если **R** – представление графов, принадлежащих множест-

ву \mathbf{T} , списками смежности вершин, то задача проверки отношения $G_1 \leq G_2$? принадлежит классу $LCl(\mathbf{R})$, что и требовалось показать.

Пусть \mathbf{R} – представление графов, принадлежащих множеству \mathbf{T} , списками смежности вершин. Выделим из матрицы \mathbf{M}_{G_2} смежности вершин графа G_2 матрицу \mathbf{M} порядка $|V_1| \times |V_1|$, расположенную в верхнем левом углу. Ясно, что матрица \mathbf{M} представляет подграф G графа G_2 , индуцированный множеством вершин V_1 . Теперь для матрицы \mathbf{M}_{G_1} смежности вершин графа G_1 и для матрицы \mathbf{M} достаточно проверить истинность неравенства $\mathbf{M}_{G_1} \leq \mathbf{M}$. Ясно, что если это неравенство – истинное, то $G_1 \leq G_2$, иначе $G_1 \not\leq G_2$. Все описанные действия принадлежат классу $LCl(\mathbf{R})$, где \mathbf{R} – представление графов, принадлежащих множеству \mathbf{T} , матрицами смежности вершин. Следовательно, если \mathbf{R} – представление графов, принадлежащих множеству \mathbf{T} , матрицами смежности вершин, то задача проверки отношения $G_1 \leq G_2$? принадлежит классу $LCl(\mathbf{R})$, что и требовалось показать.

Пусть \mathbf{R} – представление графов, принадлежащих множеству \mathbf{T} , либо списками, либо матрицами, смежности ребер. С помощью таблиц 1.1 графов G_1 и G_2 выделим такое подмножество J множества E_2 , что $j \in J$ тогда и только тогда, когда концы ребра j графа G_2 – это элементы множества V_1 , смежные в графе G_1 . Ясно, что если $|J| < |E_1|$, то $G_1 \not\leq G_2$, а если $|J| = |E_1|$, то $G_1 \leq G_2$. Описанные действия принадлежат классу $LCl(\mathbf{R})$, где \mathbf{R} – представление графов, принадлежащих множеству \mathbf{T} , либо списками, либо матрицами, смежности вершин. Следовательно, если \mathbf{R} – представление графов, принадлежащих множеству \mathbf{T} , либо списками, либо матрицами смежности ребер, то задача проверки отношения $G_1 \leq G_2$? принадлежит классу $LCl(\mathbf{R})$, что и требовалось показать.

2. Рассмотрим задачу проверки отношения $G_1 \triangleleft G_2$?. Пусть $f \in S(|V_2|)$. Обозначим через $f(G_2)$ граф, полученный в результате изменения нумерации вершин графа G_2 в соответствии с правилом: $i \rightarrow f(i)$ ($i \in V_2$). Ясно, что

$$G_1 \triangleleft G_2 \Leftrightarrow (\exists f \in S(|V_2|))(G_1 \leq f(G_2)).$$

Как применение процедуры $PRMTTNS_2(a_1 \dots a_{|V_2|})$ (см. пример 1.9), порождающей очередную подстановку $f \in S(|V_2|)$, так и построение графа $f(G_2)$ не выводят за пределы класса $LCl(\mathbf{R})$, где \mathbf{R} – представление графов, принадлежащих множеству \mathbf{T} , либо списками, либо матрицами смежности (либо вершин, либо ребер). Кроме того, как было показано выше, проверка отношения $G_1 \leq f(G_2)$? также не выводит за пределы класса

$LCI(\mathbf{R})$, где \mathbf{R} – представление графов, принадлежащих множеству \mathbf{T} , либо списками, либо матрицами смежности (либо вершин, либо ребер). Следовательно, задача проверки истинности высказывания

$$(\exists f \in S(|V_2|))(G_1 \leq f(G_2))$$

принадлежит классу $LCI(\mathbf{R})$, где \mathbf{R} – представление графов, принадлежащих множеству \mathbf{T} , либо списками, либо матрицами смежности (либо вершин, либо ребер), что и требовалось показать.

Теорема доказана.

Замечание 2.8. Из теоремы 2.6 вытекает, что в классе $LCI(\mathbf{R})$ (где \mathbf{R} – представление графов, принадлежащих множеству \mathbf{T} , либо списками, либо матрицами смежности (либо вершин, либо ребер)) содержатся задачи, решение которых сводится к последовательности проверок для исходного графа $G = (V, E) \in \mathbf{T}$ отношений

$$G^{(i)} \triangleleft G \quad (i = 1, \dots, l),$$

где $G^{(i)} \in \mathbf{T}_{|V|}$ ($i = 1, \dots, l$), причем, либо l – фиксированное число, либо значение l зависит от исходного графа G , но представление графа $G^{(i+1)}$ восстанавливается по представлениям графов $G^{(i)}, \dots, G^{(i-h)}$, где h – константа, в рамках памяти $O(v(\mathbf{R}(G)))$ ($|V| \rightarrow \infty, |E| \rightarrow \infty$). Проиллюстрируем сказанное рядом примеров.

Пример 2.1. Пусть \mathbf{R} – представление графов, принадлежащих множеству \mathbf{T} , либо списками, либо матрицами смежности (либо вершин, либо ребер). Тогда:

1. Классу $LCI(\mathbf{R})$ принадлежит задача проверки для графа $G = (V, E) \in \mathbf{T}$ свойства “быть двудольным графом” (см. пример 1.10).

Действительно (см. теорему 1.10), граф G – двудольный тогда и только тогда, когда G не содержит циклов нечетной длины. Поэтому достаточно проверить истинность высказывания

$$(\exists i \in \{2, \dots, \lfloor 0.5 \cdot (|V| + 1) \rfloor\})(C_{2i-1} \triangleleft G),$$

где C_{2i-1} ($i \in \{2, \dots, \lfloor 0.5 \cdot (|V| + 1) \rfloor\}$) – граф, изображенный на рис. 2.3.

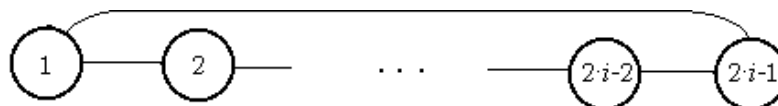


Рис. 2.3. Граф C_{2i-1} .

В данном случае $l = \lfloor 0.5 \cdot (|V| + 1) \rfloor - 1$, т.е. значение l зависит от исходного графа G . Однако, представление графа

$$C_{2^{(i+1)-1}} = C_{2^{i+1}} \quad (i \in \{2, \dots, \lfloor 0.5 \cdot (|V| + 1) \rfloor - 1\})$$

восстанавливается по представлению графа $C_{2^{i+1}}$ ($i \in \{2, \dots, \lfloor 0.5 \cdot (|V| + 1) \rfloor - 1\}$) в рамках памяти $\nu(\mathbf{R}(C_{2^{i+1}}))$, а, следовательно, и в рамках памяти $\nu(\mathbf{R}(G))$.

2. Классу $LCl(\mathbf{R})$ принадлежит задача проверки для графа $G = (V, E) \in \mathbf{T}$ истинности равенства

$$\text{intrsctn}(G) = |E|?,$$

где $\text{intrsctn}(G)$ – число пересечения графа G , т.е. наименьшая мощность такого множества S , что граф G изоморфен некоторому графу $G(F) = (F, E(F))$, где множество вершин

$$F = \{S_1, \dots, S_{|V|}\}$$

состоит из (попарно различных) подмножеств множества S , объединение которых равно S , а множество ребер $E(F)$ определяется равенством

$$E(F) = \{\{S_i, S_j\} \subseteq F \mid (i \neq j) \& (S_i \cap S_j \neq \emptyset)\}.$$

Действительно, известно (см., напр., [12]), что равенство $\text{intrsctn}(G) = |E|$ истинно тогда и только тогда, когда граф G не содержит треугольников. Поэтому достаточно проверить истинность отношения $C_3 \triangleleft G$. В данном случае $l = 1$, т.е. l – фиксированное число.

2. Классу $LCl(\mathbf{R})$ принадлежит задача проверки для графа $G = (V, E) \in \mathbf{T}$ истинности высказывания

$$(\exists X \in \mathbf{T})(L(X) \sim G), \quad (2.88)$$

где \sim – отношение изоморфизма графов, принадлежащих множеству \mathbf{T} .

Действительно, известно (см., напр., [12]), что высказывание (2.88) – истинное тогда и только тогда, когда ни один из графов $G^{(i)}$ ($i = 1, \dots, 9$) (см. рис. 2.4) не изоморфен ни одному индуцированному подграфу графа G . Поэтому достаточно последовательно проверить истинность отношений

$$G^{(i)} \triangleleft G \quad (i = 1, \dots, 9).$$

При этом для каждого такого $i \in \{1, \dots, 9\}$, что $G^{(i)} \triangleleft G$, проверяется, является ли соответствующий подграф графа G индуцированным подграфом (очевидно, что последняя проверка является задачей, принадлежащей классу $LCl(\mathbf{R})$, если \mathbf{R} – представление графов, принадлежащих множеству \mathbf{T} , либо списками, либо матрицами смежности (либо вершин, либо ребер)). В данном случае $l = 9$, т.е. l – фиксированное число.

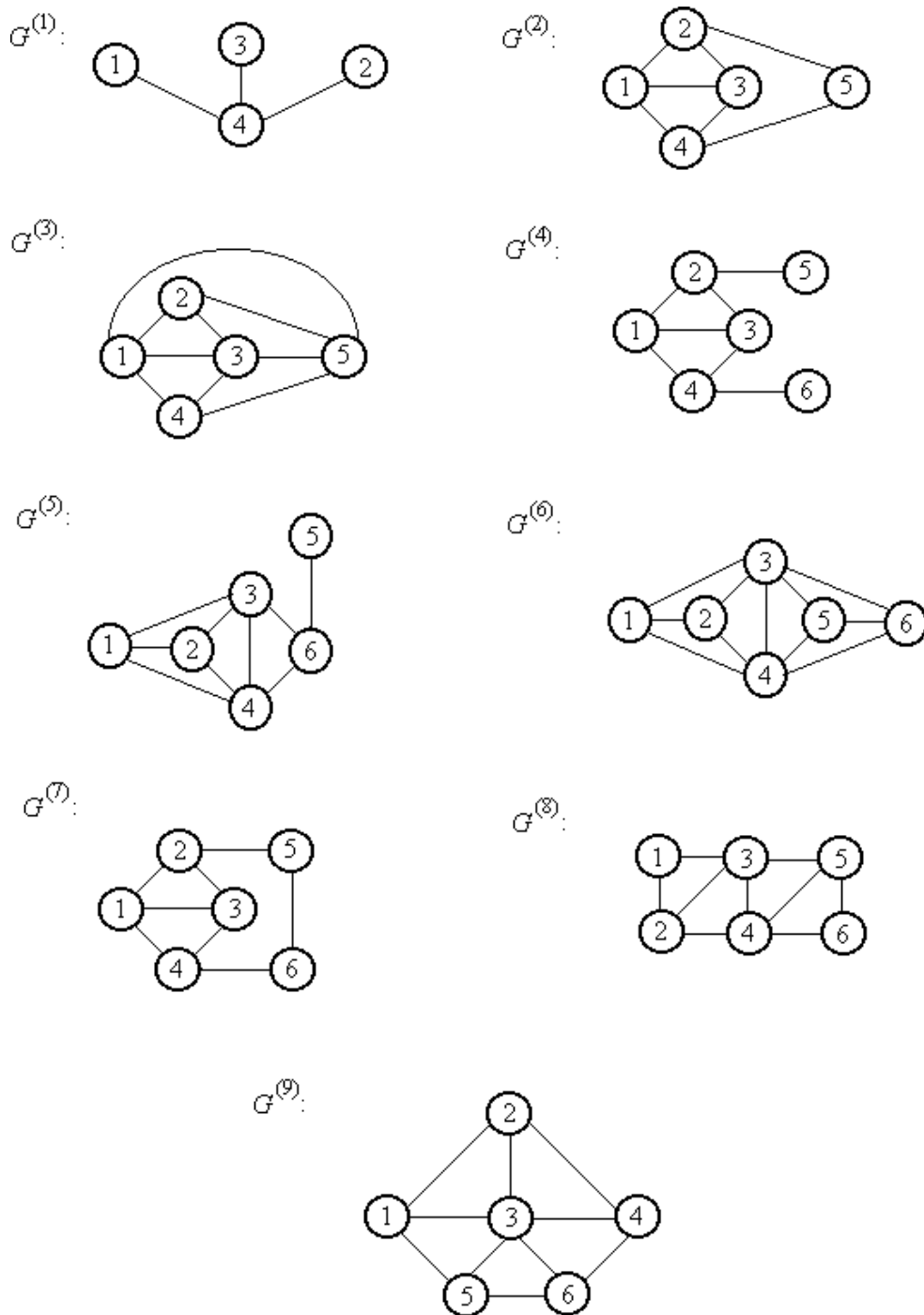


Рис. 2.4. Запрещенные подграфы для реберного графа.

2.3. Представления направленных графов списками смежности.

Для каждого $n \in \mathbf{N}$ обозначим через $\overrightarrow{\mathbf{G}}^{or}(n, m)$ и $\overrightarrow{\mathbf{G}}^{dir}(n, m)$ множество во всех, соответственно, орграфов и направленных графов $G = (V, E)$ ($E \subseteq (V \times V) \setminus \{(v, v) \mid v \in V\}$) с множеством вершин $V = \{1, \dots, n\}$ и m дугами. Из определений орграфа и направленного графа непосредственно вытекает, что для орграфа

$$m \in \{0, 1, \dots, \binom{n}{2}\} \quad (n \in \mathbf{N})$$

а для направленного графа

$$m \in \{0, 1, \dots, n^2 - n\} \quad (n \in \mathbf{N}).$$

Ясно, что

$$\overrightarrow{\mathbf{G}}^{or}(n, m) \subseteq \overrightarrow{\mathbf{G}}^{dir}(n, m)$$

для всех $n \in \mathbf{N}$ и $m \in \{0, 1, \dots, n^2 - n\}$.

По аналогии с обычными графами считаем, что $E = \{1, \dots, m\}$ в случае представления $G \in \overrightarrow{\mathbf{G}}^{dir}(n, m)$ матрицами или списками смежности дуг, причем, как и для обычных графов, задана таблица, определяющая нумерацию дуг G (эту таблицу также назовем таблицей 1.1).

Определения классов $Lcl(\mathbf{R})$ и $WORK-Lcl(\mathbf{R})$ переносятся на множества $\overrightarrow{\mathbf{G}}^{or}(n, m)$ и $\overrightarrow{\mathbf{G}}^{dir}(n, m)$ естественным образом.

Представления $G \in \overrightarrow{\mathbf{G}}^{dir}(n, m)$ списками смежности вершин и списками смежности дуг, в отличие от обычных графов, имеют различную структуру. Поэтому рассмотрим их по отдельности.

В случае представления $G = (V, E) \in \overrightarrow{\mathbf{G}}^{dir}(n, m)$ списками смежности вершин, для каждой вершины $v \in V$ необходимо выделить следующие две ее открытые окрестности

$$\Gamma^+(v) = \{v' \in V \mid (v, v') \in E\} \quad (v \in V),$$

$$\Gamma^-(v) = \{v' \in V \mid (v', v) \in E\} \quad (v \in V).$$

Замечание 2.9. 1. Введенные выше открытые окрестности вершины характеризуют следующие типы вершин $G = (V, E) \in \overrightarrow{\mathbf{G}}^{dir}(n, m)$:

а) $v \in V$ – источник (иногда такую вершину называют *граничной*) тогда и только тогда, когда $\Gamma^+(v) \neq \emptyset$ и $\Gamma^-(v) = \emptyset$;

б) $v \in V$ – *сток* (в ориентированном дереве такую вершину называют *висячей* или *листом*) тогда и только тогда, когда $\Gamma^+(v) = \emptyset$ и $\Gamma^-(v) \neq \emptyset$;

в) $v \in V$ – *изолированная вершина* тогда и только тогда, когда $\Gamma^+(v) \cup \Gamma^-(v) = \emptyset$.

2. Из определения орграфа вытекает, что:

а) $G = (V, E) \in \overrightarrow{\mathbf{G}}^{or}(n, m) \Rightarrow (\forall v \in V)(\Gamma^+(v) \cap \Gamma^-(v) = \emptyset)$;

б) число различных орграфов, которые могут быть получены в результате ориентации ребер графа $G = (V, E) \in \mathbf{T}$, равно $2^{|E|}$.

Определение 2.2. Представлением $G = (V, E) \in \overrightarrow{\mathbf{G}}^{dir}(n, m)$ списками смежности вершин назовем упорядоченную пару

$$(\mathbf{LIST}^+, \mathbf{LIST}^-)$$

n -элементных последовательностей списков

$$\mathbf{LIST}^+ : \mathbf{LIST}_1^+, \mathbf{LIST}_2^+, \dots, \mathbf{LIST}_n^+, \quad (2.89)$$

$$\mathbf{LIST}^- : \mathbf{LIST}_1^-, \mathbf{LIST}_2^-, \dots, \mathbf{LIST}_n^-, \quad (2.90)$$

где элементы списка \mathbf{LIST}_i^+ ($i = 1, \dots, n$) – это элементы множества $\Gamma^+(i)$, а элементы списка \mathbf{LIST}_i^- ($i = 1, \dots, n$) – это элементы множества $\Gamma^-(i)$.

В случае представления $G = (V, E) \in \overrightarrow{\mathbf{G}}^{dir}(n, m)$ списками смежности дуг, для каждой дуги $\vec{e} = (v, v') \in E$ необходимо выделить следующие четыре ее открытые окрестности

$$\Gamma_b^+(\vec{e}) = \{(v, v'') \in E \mid v'' \in V \setminus \{v'\}\},$$

$$\Gamma_b^-(\vec{e}) = \{(v'', v) \in E \mid v'' \in V\},$$

$$\Gamma_e^+(\vec{e}) = \{(v', v'') \in E \mid v'' \in V\},$$

$$\Gamma_e^-(\vec{e}) = \{(v'', v') \in E \mid v'' \in V \setminus \{v\}\}.$$

Замечание 2.10. Из определения открытых окрестностей дуг непосредственно вытекает, что для любого $G = (V, E) \in \overrightarrow{\mathbf{G}}^{dir}(n, m)$ истинными являются следующие утверждения:

$$1) (\forall \vec{e} \in E)(\Gamma_b^+(\vec{e}) \cap \Gamma_e^-(\vec{e}) = \emptyset);$$

$$2) (\forall \vec{e} \in E)(\Gamma_e^+(\vec{e}) \cap \Gamma_b^-(\vec{e}) = \emptyset) \Rightarrow G \in \overrightarrow{\mathbf{G}}^{or}(n, m);$$

3) если $\vec{e} = (v, v') \in E$ – такая дуга, что

$$(\Gamma_b^+(\vec{e}) \cup \Gamma_b^-(\vec{e}) = \emptyset) \vee (\Gamma_e^+(\vec{e}) \cup \Gamma_e^-(\vec{e}) = \emptyset),$$

то \vec{e} – мост, причем, $v \in V$ – источник, если $\Gamma_b^+(\vec{e}) \cup \Gamma_b^-(\vec{e}) = \emptyset$, и $v' \in V$ – сток если $\Gamma_e^+(\vec{e}) \cup \Gamma_e^-(\vec{e}) = \emptyset$.

Определение 2.3. Представлением $G = (V, E) \in \overrightarrow{\mathbf{G}}^{dir}(n, m)$ списками смежности дуг назовем упорядоченную четверку

$$(LIST_b^+, LIST_b^-, LIST_e^+, LIST_e^-)$$

m -элементных последовательностей списков

$$LIST_b^+ : LIST_{b1}^+, LIST_{b2}^+, \dots, LIST_{bm}^+, \quad (2.91)$$

$$LIST_b^- : LIST_{b1}^-, LIST_{b2}^-, \dots, LIST_{bm}^-, \quad (2.92)$$

$$LIST_e^+ : LIST_{e1}^+, LIST_{e2}^+, \dots, LIST_{em}^+, \quad (2.93)$$

$$LIST_e^- : LIST_{e1}^-, LIST_{e2}^-, \dots, LIST_{em}^-, \quad (2.94)$$

где элементы списка $LIST_{bi}^+$ ($i = 1, \dots, m$) – это элементы множества $\Gamma_b^+(i)$, элементы списка $LIST_{bi}^-$ ($i = 1, \dots, m$) – это элементы множества $\Gamma_b^-(i)$, элементы списка $LIST_{ei}^+$ ($i = 1, \dots, m$) – это элементы множества $\Gamma_e^+(i)$, а элементы списка $LIST_{ei}^-$ ($i = 1, \dots, m$) – это элементы множества $\Gamma_e^-(i)$.

Пример 2.2. 1. Рассмотрим направленный граф $G_2 \in \overrightarrow{\mathbf{G}}^{dir}(6, 9)$ (рис. 2.5 а)).

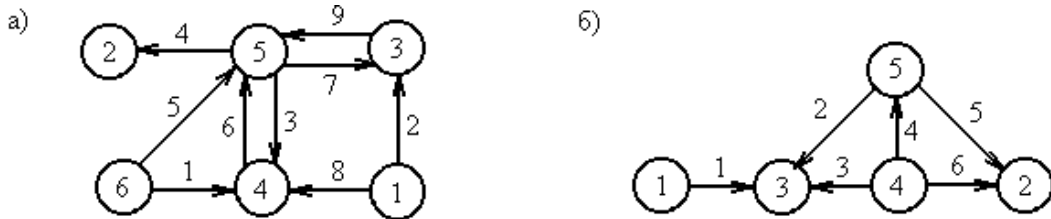


Рис. 2.5. Графическое изображение: а) направленного графа G_2 ; б) оргафа G_3 .

Вычислим открытые окрестности вершин направленного графа G_2

$$\begin{aligned}\Gamma^+(1) &= \{3,4\}, \quad \Gamma^-(1) = \Gamma^+(2) = \Gamma^-(6) = \emptyset, \\ \Gamma^-(2) &= \Gamma^+(3) = \Gamma^+(4) = \{5\}, \quad \Gamma^-(3) = \{5,1\}, \quad \Gamma^-(4) = \{5,1,6\}, \\ \Gamma^+(5) &= \{2,4,3\}, \quad \Gamma^-(5) = \{6,4,3\}, \quad \Gamma^+(6) = \{5,4\}.\end{aligned}$$

Так как $\Gamma^+(1) \neq \emptyset$, $\Gamma^-(1) = \emptyset$ и $\Gamma^+(6) \neq \emptyset$, $\Gamma^-(6) = \emptyset$, то вершины 1 и 6 являются источниками, а так как $\Gamma^+(2) = \emptyset$, $\Gamma^-(2) \neq \emptyset$, то вершина 2 – сток.

Представление направленного графа G_2 списками смежности вершин изображено на рис 2.6.

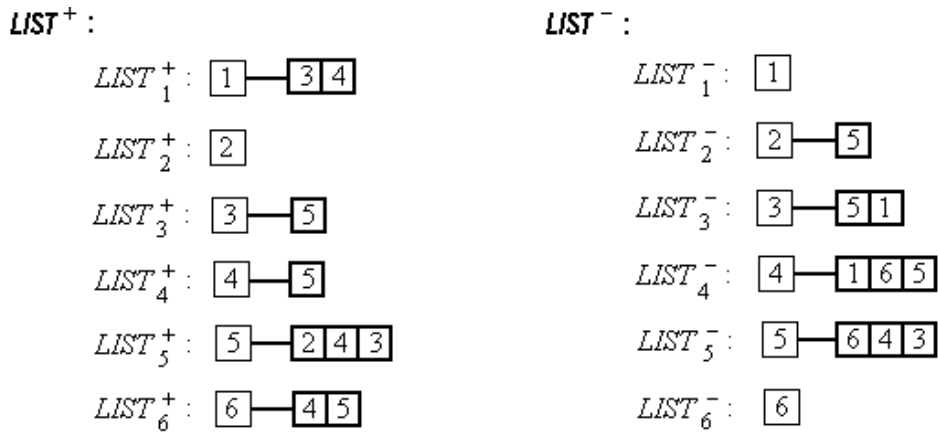


Рис. 2.6. Представление направленного графа G_2 списками смежности вершин.

Вычислим открытые окрестности дуг направленного графа G_2

$$\begin{aligned}\Gamma_b^+(1) &= \{5\}, \quad \Gamma_e^+(1) = \Gamma_e^+(3) = \Gamma_e^+(8) = \{6\}, \\ \Gamma_b^-(1) &= \Gamma_b^-(2) = \Gamma_e^+(4) = \Gamma_e^-(4) = \Gamma_b^-(5) = \Gamma_b^+(6) = \Gamma_b^-(8) = \Gamma_b^+(9) = \emptyset, \\ \Gamma_e^-(1) &= \{3,8\}, \quad \Gamma_b^+(2) = \{8\}, \quad \Gamma_e^+(2) = \Gamma_e^+(7) = \{9\}, \\ \Gamma_e^-(2) &= \{7\}, \quad \Gamma_b^+(3) = \{4,7\}, \quad \Gamma_b^-(3) = \Gamma_b^-(4) = \Gamma_b^-(7) = \{5,6,9\}, \\ \Gamma_e^-(3) &= \{1,8\}, \quad \Gamma_b^+(4) = \{7,3\}, \quad \Gamma_b^+(5) = \{1\}, \\ \Gamma_e^+(5) &= \Gamma_e^+(6) = \Gamma_e^+(9) = \{7,3,4\}, \quad \Gamma_e^-(5) = \{9,6\}, \\ \Gamma_b^-(6) &= \{3,8,1\}, \quad \Gamma_e^-(6) = \{9,5\}, \quad \Gamma_b^+(7) = \{3,4\}, \\ \Gamma_e^-(7) &= \Gamma_b^+(8) = \{2\}, \quad \Gamma_e^-(8) = \{1,3\}, \quad \Gamma_b^-(9) = \{7,2\}, \quad \Gamma_e^-(9) = \{5,6\}.\end{aligned}$$

Так как $\Gamma_e^+(4) \cup \Gamma_e^-(4) = \emptyset$, то дуга 4 – мост, причем ее конец, т.е. вершина 2, является стоком.

Представление направленного графа G_2 списками смежности дуг изображено на рис 2.7.

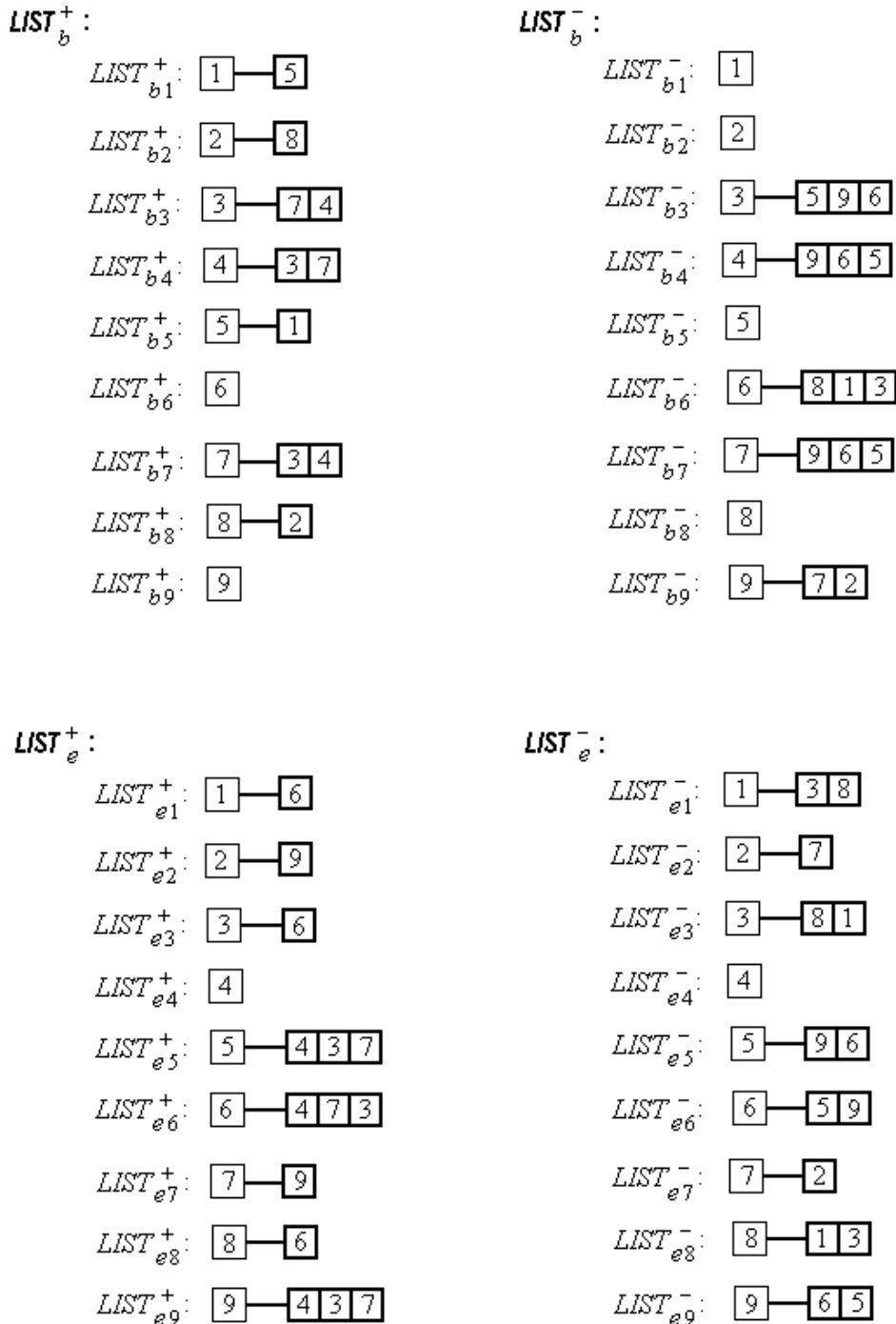


Рис. 2.7. Представление направленного графа G_2 списками смежности дуг.

2. Рассмотрим оргграф $G_3 = \vec{G}^{or}(5,6)$ (рис. 2.5 б)), полученный ориентацией ребер графа G_1 из примера 1.1.

Вычислим открытые окрестности вершин оргграфа G_3

$$\Gamma^+(1) = \{3\}, \quad \Gamma^-(1) = \Gamma^+(2) = \Gamma^+(3) = \Gamma^-(4) = \emptyset,$$

$$\Gamma^-(2) = \{4,5\}, \quad \Gamma^-(3) = \{4,5,1\}, \quad \Gamma^+(4) = \{3,2,5\},$$

$$\Gamma^+(5) = \{3,2\}, \quad \Gamma^-(5) = \{4\}.$$

Так как $\Gamma^+(1) \neq \emptyset$, $\Gamma^-(1) = \emptyset$ и $\Gamma^+(4) \neq \emptyset$, $\Gamma^-(4) = \emptyset$, то вершины 1 и 4 являются источниками, а так как $\Gamma^+(2) = \emptyset$, $\Gamma^-(2) \neq \emptyset$ и $\Gamma^+(3) = \emptyset$, $\Gamma^-(3) \neq \emptyset$, то каждая из вершин 2 и 3 является стоком.

Представление оргграфа G_3 списками смежности вершин изображено на рис 2.8.

$LIST^+$:

$$LIST_1^+ : \boxed{1} \text{---} \boxed{3}$$

$$LIST_2^+ : \boxed{2}$$

$$LIST_3^+ : \boxed{3}$$

$$LIST_4^+ : \boxed{4} \text{---} \boxed{3} \boxed{5} \boxed{2}$$

$$LIST_5^+ : \boxed{5} \text{---} \boxed{3} \boxed{2}$$

$LIST^-$:

$$LIST_1^- : \boxed{1}$$

$$LIST_2^- : \boxed{2} \text{---} \boxed{5} \boxed{4}$$

$$LIST_3^- : \boxed{3} \text{---} \boxed{1} \boxed{4} \boxed{5}$$

$$LIST_4^- : \boxed{4}$$

$$LIST_5^- : \boxed{5} \text{---} \boxed{4}$$

Рис. 2.8. Представление оргграфа G_3 списками смежности вершин.

Вычислим открытые окрестности дуг оргграфа G_3

$$\Gamma_b^+(1) = \Gamma_b^-(1) = \Gamma_e^+(1) = \Gamma_e^+(2) = \Gamma_b^-(3) = \Gamma_3^+(3) =$$

$$= \Gamma_b^-(4) = \Gamma_e^-(4) = \Gamma_e^+(5) = \Gamma_b^-(6) = \Gamma_e^+(6) = \emptyset,$$

$$\Gamma_e^-(1) = \{3,2\}, \quad \Gamma_b^+(2) = \Gamma_e^-(6) = \{5\}, \quad \Gamma_b^-(2) = \Gamma_b^-(5) = \{4\},$$

$$\Gamma_e^-(2) = \{1,3\}, \quad \Gamma_b^+(3) = \{4,6\}, \quad \Gamma_e^-(3) = \{1,2\}, \quad \Gamma_b^+(4) = \{6,3\},$$

$$\Gamma_e^+(4) = \{2,5\}, \quad \Gamma_b^+(5) = \{2\}, \quad \Gamma_e^-(5) = \{6\}, \quad \Gamma_b^+(6) = \{3,4\}.$$

Так как $\Gamma_b^+(1) \cup \Gamma_b^-(1) = \emptyset$, то дуга 1 – мост, причем ее начало, т.е. вершина 1 – источник.

Представление оргграфа G_3 списками смежности дуг изображено на рис 2.9.

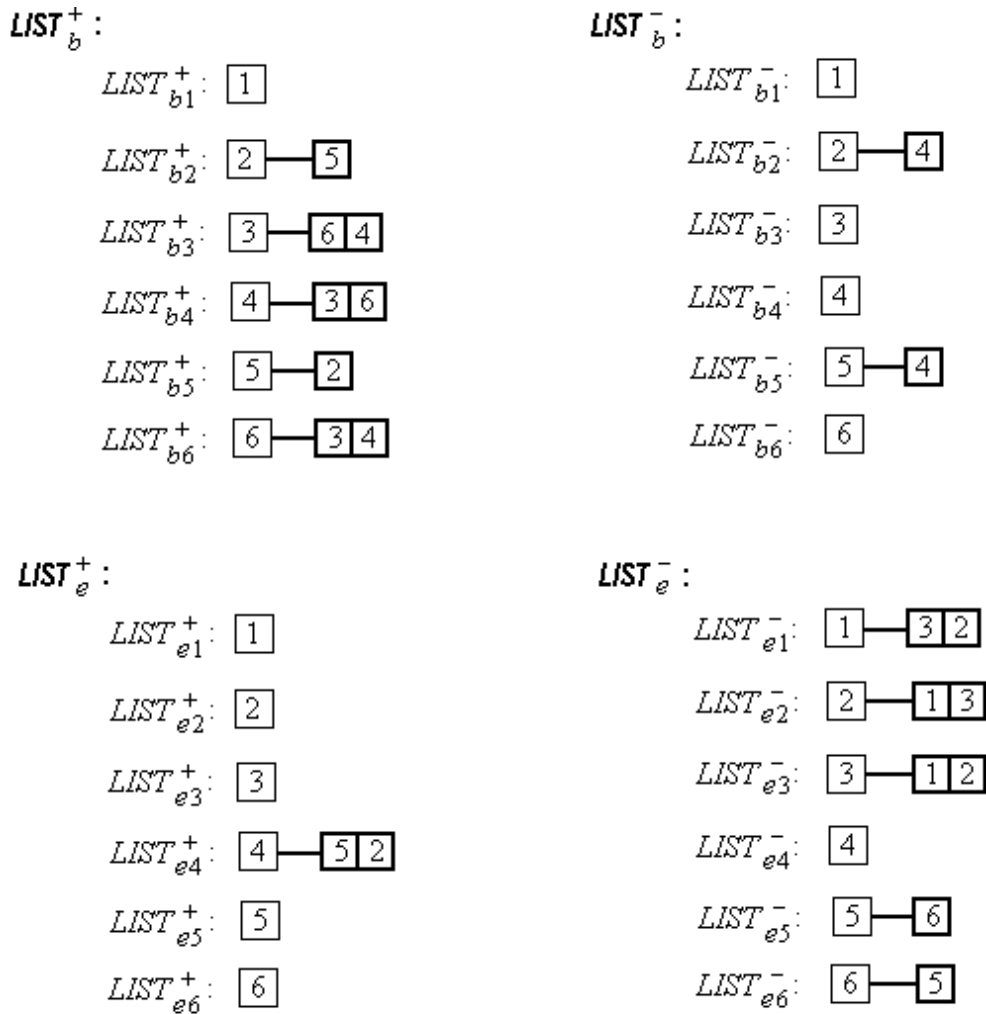


Рис. 2.9. Представление орграфа G_3 списками смежности дуг.

Принимая во внимание отмеченные в п.2.2 соображения относительно порядка следования элементов в списках, следующим образом определим понятие *стандартная форма представления списками смежности* для орграфов и направленных графов.

Определение 2.4. Назовем стандартной формой представления $G = (V, E) \in \overline{\mathbf{G}}^{dir}(n, m)$ списками смежности вершин такую упорядоченную пару $(LIST^+, LIST^-)$ n -элементных последовательностей списков (2.89) и (2.90), что элементы каждого из списков $LIST_i^+, LIST_i^-$ ($i = 1, \dots, n$) расположены в порядке их возрастания.

Определение 2.5. Назовем стандартной формой представления $G = (V, E) \in \overrightarrow{\mathbf{G}}^{dir}(n, m)$ списками смежности дуг такую упорядоченную четверку $(LIST_b^+, LIST_b^-, LIST_e^+, LIST_e^-)$ m -элементных последовательностей списков (2.91)-(2.94), что элементы каждого из списков $LIST_{bi}^+$, $LIST_{bi}^-$, $LIST_{ei}^+$ и $LIST_{ei}^-$ ($i = 1, \dots, m$) расположены в порядке их возрастания.

Положим

$$\overrightarrow{\mathbf{S}}_n^{or} = \bigcup_{m=0}^{\binom{n}{2}} \overrightarrow{\mathbf{G}}^{or}(n, m),$$

$$\overrightarrow{\mathbf{S}}_n^{dir} = \bigcup_{m=0}^{n \cdot (n-1)} \overrightarrow{\mathbf{G}}^{dir}(n, m).$$

Операции и отношения алгебраической системы \mathbf{A}_n естественным образом могут быть перенесены на множество $\overrightarrow{\mathbf{S}}_n^{dir}$, а операции и отношения алгебраической системы \mathbf{B} – на множество $\overrightarrow{\mathbf{T}} = \bigcup_{n=1}^{\infty} \overrightarrow{\mathbf{T}}_n$, где $\overrightarrow{\mathbf{T}}_n = \bigcup_{k=1}^n \overrightarrow{\mathbf{S}}_k^{dir}$. Исследуем с этой точки зрения сложность представления орграфов и направленных графов списками смежности.

Вычислим вначале мощности множеств $\overrightarrow{\mathbf{S}}_n^{or}$ и $\overrightarrow{\mathbf{S}}_n^{dir}$.

Утверждение 2.1. Для всех $n \in \mathbf{N}$

$$|\overrightarrow{\mathbf{S}}_n^{or}| = 3^{\binom{n}{2}}, \quad (2.95)$$

$$|\overrightarrow{\mathbf{S}}_n^{dir}| = 2^{n \cdot (n-1)}. \quad (2.96)$$

Доказательство. Так как орграфы, полученные в результате ориентации различных графов $G \in \mathbf{S}_n$ – различные, то

$$|\overrightarrow{\mathbf{S}}_n^{or}| = \sum_{m=0}^{\binom{n}{2}} |\overrightarrow{\mathbf{G}}^{or}(n, m)| = \sum_{m=0}^{\binom{n}{2}} \binom{\binom{n}{2}}{m} \cdot 2^m. \quad (2.97)$$

Положим $l = \binom{n}{2}$ и $x = 2$ в тождестве

$$(1+x)^l = \sum_{m=0}^l \binom{l}{m} \cdot x^m. \quad (2.98)$$

Получим

$$\sum_{m=0}^{\binom{n}{2}} \binom{\binom{n}{2}}{m} \cdot 2^m = 3^{\binom{n}{2}}. \quad (2.99)$$

Подставим (2.99) в (2.97). Получим (2.95), что и требовалось показать.

Так как

$$m_1 \neq m_2 \Rightarrow \overrightarrow{\mathbf{G}}^{dir}(n, m_1) \cap \overrightarrow{\mathbf{G}}^{dir}(n, m_2) = \emptyset,$$

то

$$|\overrightarrow{\mathbf{S}}_n^{dir}| = \sum_{m=0}^{n \cdot (n-1)} |\overrightarrow{\mathbf{G}}^{dir}(n, m)| = \sum_{m=0}^{n \cdot (n-1)} \binom{n \cdot (n-1)}{m}. \quad (2.100)$$

Положим $l = n \cdot (n-1)$ и $x = 1$ в (2.98). Получим

$$\sum_{m=0}^{n \cdot (n-1)} \binom{n \cdot (n-1)}{m} = 2^{n \cdot (n-1)}. \quad (2.101)$$

Подставим (2.101) в (2.100). Получим (2.96), что и требовалось доказать.

Утверждение доказано.

Теперь (применяя технику, использованную при доказательстве теорем 2.2 и 2.3) вычислим типичные значения дуг орграфов и направленных графов.

Теорема 2.7. Среднее число дуг орграфа, случайно выбранного из множества $\overrightarrow{\mathbf{S}}_n^{or}$, равно $\frac{2}{3} \cdot \binom{n}{2}$.

Доказательство. Рассмотрим случайную величину ξ_n , принимающую значение m ($m \in \{0, 1, \dots, \binom{n}{2}\}$) с вероятностью

$$p_{n,m} = \frac{\binom{\binom{n}{2}}{m} \cdot 2^m}{3^{\binom{n}{2}}},$$

т.е. $p_{n,m}$ ($m \in \{0, 1, \dots, \binom{n}{2}\}$) есть вероятность того, что случайно выбранный из множества $\overrightarrow{\mathbf{S}}_n^{or}$ орграф содержит m дуг. Найдем математическое ожидание случайной величины ξ_n

$$\mathbf{M}\xi_n = \sum_{m=0}^{\binom{n}{2}} m \cdot p_{n,m} = \frac{1}{3^{\binom{n}{2}}} \cdot \sum_{m=0}^{\binom{n}{2}} m \cdot \binom{\binom{n}{2}}{m} \cdot 2^m = \frac{2}{3^{\binom{n}{2}}} \cdot \sum_{m=1}^{\binom{n}{2}} m \cdot \binom{\binom{n}{2}}{m} \cdot 2^{m-1}. \quad (2.102)$$

Продифференцируем тождество (2.98). Получим

$$l \cdot (1+x)^{l-1} = \sum_{m=1}^l \binom{l}{m} \cdot m \cdot x^{m-1}. \quad (2.103)$$

Положим в (2.103) $l = \binom{n}{2}$ и $x = 2$. Получим

$$\sum_{m=1}^{\binom{n}{2}} m \cdot \binom{\binom{n}{2}}{m} \cdot 2^{m-1} = \binom{n}{2} \cdot 3^{\binom{n}{2}-1}. \quad (2.104)$$

Подставим (2.104) в (2.102). Получим

$$\mathbf{M}\xi_n = \frac{2}{3} \cdot \binom{n}{2}, \quad (2.105)$$

что и требовалось доказать.

Теорема доказана.

Теорема 2.8. Число m дуг почти всех орграфов, принадлежащих множеству $\overrightarrow{\mathbf{S}}_n$, удовлетворяет асимптотическому равенству

$$m = \Theta(n^2) \quad (n \rightarrow \infty). \quad (2.106)$$

Доказательство. Вычислим дисперсию случайной величины ξ_n , введенной в теореме 2.7. Воспользуемся формулой

$$\mathbf{D}\xi_n = \mathbf{M}\xi_n^2 - (\mathbf{M}\xi_n)^2. \quad (2.107)$$

Вычислим $\mathbf{M}\xi_n^2$

$$\begin{aligned} \mathbf{M}\xi_n^2 &= \sum_{m=0}^{\binom{n}{2}} m^2 \cdot p_{n,m} = \frac{1}{3^{\binom{n}{2}}} \cdot \sum_{m=0}^{\binom{n}{2}} m^2 \cdot \binom{\binom{n}{2}}{m} \cdot 2^m = \frac{2}{3^{\binom{n}{2}}} \cdot \sum_{m=1}^{\binom{n}{2}} m^2 \cdot \binom{\binom{n}{2}}{m} \cdot 2^{m-1} = \\ &= \frac{2}{3^{\binom{n}{2}}} \cdot \sum_{m=1}^{\binom{n}{2}} (m \cdot (m-1) + m) \cdot \binom{\binom{n}{2}}{m} \cdot 2^{m-1} = \end{aligned}$$

$$\begin{aligned}
&= \frac{4}{3^{\binom{n}{2}}} \cdot \sum_{m=2}^{\binom{n}{2}} m \cdot (m-1) \cdot \binom{\binom{n}{2}}{m} \cdot 2^{m-2} + \frac{2}{3^{\binom{n}{2}}} \cdot \sum_{m=1}^{\binom{n}{2}} m \cdot \binom{\binom{n}{2}}{m} \cdot 2^{m-1} = \\
&= \frac{4}{3^{\binom{n}{2}}} \cdot \sum_{m=2}^{\binom{n}{2}} m \cdot (m-1) \cdot \binom{\binom{n}{2}}{m} \cdot 2^{m-2} + \mathbf{M}\xi_n. \tag{2.108}
\end{aligned}$$

Продифференцируем тождество (2.103). Получим

$$l \cdot (l-1) \cdot (1+x)^{l-2} = \sum_{m=2}^l \binom{l}{m} \cdot m \cdot (m-1) \cdot x^{m-2}. \tag{2.109}$$

Положим в (2.109) $l = \binom{n}{2}$ и $x = 2$. Получим

$$\sum_{m=2}^{\binom{n}{2}} m \cdot (m-1) \cdot \binom{\binom{n}{2}}{m} \cdot 2^{m-2} = \binom{n}{2} \cdot \left(\binom{n}{2} - 1\right) \cdot 3^{\binom{n}{2}-2}. \tag{2.110}$$

Подставим (2.110) и (2.105) в (2.108). Получим

$$\begin{aligned}
\mathbf{M}\xi_n^2 &= \frac{4}{9} \cdot \binom{n}{2} \cdot \left(\binom{n}{2} - 1\right) + \frac{2}{3} \cdot \binom{n}{2} = \\
&= \frac{4}{9} \cdot \binom{n}{2}^2 + \frac{2}{9} \cdot \binom{n}{2}. \tag{2.111}
\end{aligned}$$

Подставим (2.111) и (2.105) в (2.107). Получим

$$\mathbf{D}\xi_n = \frac{4}{9} \cdot \binom{n}{2}^2 + \frac{2}{9} \cdot \binom{n}{2} - \frac{4}{9} \cdot \binom{n}{2}^2 = \frac{2}{9} \cdot \binom{n}{2}. \tag{2.112}$$

Применим к случайной величине ξ_n неравенство Чебышева

$$P\{|\xi_n - \mathbf{M}\xi_n| \geq t\} \leq \frac{\mathbf{D}\xi_n}{t^2}. \tag{2.113}$$

Положим

$$\frac{\mathbf{D}\xi_n}{t^2} = \varepsilon. \tag{2.114}$$

По определению случайной величины ξ_n , левая часть неравенства (2.113) равна доле орграфов, принадлежащих множеству $\overline{\mathbf{S}}_n^{or}$, для которых число дуг m удовлетворяет неравенству

$$|m - \frac{2}{3} \cdot \binom{n}{2}| \geq t. \quad (2.115)$$

Следовательно, доля оргграфов, принадлежащих множеству $\overrightarrow{\mathbf{S}}_n^{or}$, для которых число дуг m удовлетворяет неравенству

$$|m - \frac{2}{3} \cdot \binom{n}{2}| < t,$$

или, что эквивалентно, неравенствам

$$\frac{2}{3} \cdot \binom{n}{2} - t < m < \frac{2}{3} \cdot \binom{n}{2} + t, \quad (2.116)$$

не меньше, чем

$$E(\varepsilon) = 1 - \varepsilon. \quad (2.117)$$

Из равенства (2.114), с учетом равенства (2.112), находим

$$t = \frac{\sqrt{2}}{3} \cdot \frac{1}{\sqrt{\varepsilon}} \cdot \sqrt{\binom{n}{2}}. \quad (2.118)$$

Подставим (2.118) в (2.116). Получаем, что доля оргграфов, принадлежащих множеству $\overrightarrow{\mathbf{S}}_n^{or}$, для которых число дуг m удовлетворяет неравенствам

$$\frac{2}{3} \cdot \binom{n}{2} - \frac{\sqrt{2}}{3} \cdot \frac{1}{\sqrt{\varepsilon}} \cdot \sqrt{\binom{n}{2}} < m < \frac{2}{3} \cdot \binom{n}{2} + \frac{\sqrt{2}}{3} \cdot \frac{1}{\sqrt{\varepsilon}} \cdot \sqrt{\binom{n}{2}}, \quad (2.119)$$

не меньше, чем

$$E(\varepsilon) = 1 - \varepsilon. \quad (2.120)$$

Положим

$$\varepsilon = \Theta(n^{-2\alpha}) \quad (n \rightarrow \infty), \quad (2.121)$$

где

$$\alpha \in (0;1). \quad (2.122)$$

Из (2.121) вытекает, что

$$\frac{\sqrt{2}}{3} \cdot \frac{1}{\sqrt{\varepsilon}} \cdot \sqrt{\binom{n}{2}} = \Theta(n^{1+\alpha}) \quad (n \rightarrow \infty), \quad (2.123)$$

а из (2.120), (2.121) и (2.122) – что

$$E(\varepsilon) \rightarrow 1 \quad (n \rightarrow \infty). \quad (2.124)$$

Следовательно, из (2.119), (2.123) и (2.124) вытекает, что число m дуг почти всех орграфов, принадлежащих множеству $\overrightarrow{\mathbf{S}}_n^{or}$, удовлетворяет асимптотическому равенству

$$m = \Theta(n^2) \quad (n \rightarrow \infty),$$

что и требовалось доказать.

Теорема доказана.

Теорема 2.9. Среднее число дуг направленного графа, случайно выбранного из множества $\overrightarrow{\mathbf{S}}_n^{dir}$, равно $0.5 \cdot n \cdot (n-1)$.

Доказательство. Рассмотрим случайную величину ξ_n , принимающую значение m ($m \in \{0, 1, \dots, n \cdot (n-1)\}$) с вероятностью

$$p_{n,m} = \frac{\binom{n \cdot (n-1)}{m}}{2^{n \cdot (n-1)}},$$

т.е. $p_{n,m}$ ($m \in \{0, 1, \dots, n \cdot (n-1)\}$) есть вероятность того, что случайно выбранный из множества $\overrightarrow{\mathbf{S}}_n^{dir}$ направленный граф содержит m дуг. Найдем математическое ожидание случайной величины ξ_n

$$\begin{aligned} \mathbf{M}\xi_n &= \sum_{m=0}^{n \cdot (n-1)} m \cdot p_{n,m} = \frac{1}{2^{n \cdot (n-1)}} \cdot \sum_{m=0}^{n \cdot (n-1)} m \cdot \binom{n \cdot (n-1)}{m} = \\ &= \frac{1}{2^{n \cdot (n-1)}} \cdot \sum_{m=1}^{n \cdot (n-1)} m \cdot \binom{n \cdot (n-1)}{m} \end{aligned} \quad (2.125)$$

Положим в (2.103) $l = n \cdot (n-1)$ и $x = 1$. Получим

$$\sum_{m=1}^{n \cdot (n-1)} m \cdot \binom{n \cdot (n-1)}{m} = n \cdot (n-1) \cdot 2^{n \cdot (n-1)-1}. \quad (2.126)$$

Подставим (2.126) в (2.125). Получим

$$\mathbf{M}\xi_n = 0.5 \cdot n \cdot (n-1), \quad (2.127)$$

что и требовалось доказать.

Теорема доказана.

Теорема 2.10. Число m дуг почти всех направленных графов, принадлежащих множеству $\overrightarrow{\mathbf{S}}_n^{dir}$, удовлетворяет асимптотическому равенству

$$m = \Theta(n^2) \quad (n \rightarrow \infty). \quad (2.128)$$

Доказательство. Вычислим дисперсию случайной величины ξ_n , введенной в теореме 2.9. Воспользуемся формулой

$$\mathbf{D}\xi_n = \mathbf{M}\xi_n^2 - (\mathbf{M}\xi_n)^2. \quad (2.129)$$

Вычислим $\mathbf{M}\xi_n^2$

$$\begin{aligned} \mathbf{M}\xi_n^2 &= \sum_{m=0}^{n \cdot (n-1)} m^2 \cdot p_{n,m} = \frac{1}{2^{n \cdot (n-1)}} \cdot \sum_{m=0}^{n \cdot (n-1)} m^2 \cdot \binom{n \cdot (n-1)}{m} = \\ &= \frac{1}{2^{n \cdot (n-1)}} \cdot \sum_{m=1}^{n \cdot (n-1)} m^2 \cdot \binom{n \cdot (n-1)}{m} = \frac{1}{2^{n \cdot (n-1)}} \cdot \sum_{m=1}^{n \cdot (n-1)} (m \cdot (m-1) + m) \cdot \binom{n \cdot (n-1)}{m} = \\ &= \frac{1}{2^{n \cdot (n-1)}} \cdot \sum_{m=2}^{n \cdot (n-1)} m \cdot (m-1) \cdot \binom{n \cdot (n-1)}{m} + \frac{1}{2^{n \cdot (n-1)}} \cdot \sum_{m=1}^{n \cdot (n-1)} m \cdot \binom{n \cdot (n-1)}{m} = \\ &= \frac{1}{2^{n \cdot (n-1)}} \cdot \sum_{m=2}^{n \cdot (n-1)} m \cdot (m-1) \cdot \binom{n \cdot (n-1)}{m} + \mathbf{M}\xi_n. \end{aligned} \quad (2.130)$$

Положим в (2.109) $l = n \cdot (n-1)$ и $x = 1$. Получим

$$\sum_{m=2}^{n \cdot (n-1)} m \cdot (m-1) \cdot \binom{n \cdot (n-1)}{m} = n \cdot (n-1) \cdot (n \cdot (n-1) - 1) \cdot 2^{n \cdot (n-1) - 2}. \quad (2.131)$$

Подставим (2.131) и (2.127) в (2.130). Получим

$$\begin{aligned} \mathbf{M}\xi_n^2 &= 0.25 \cdot n \cdot (n-1) \cdot (n \cdot (n-1) - 1) + 0.5 \cdot n \cdot (n-1) = \\ &= 0.25 \cdot n^2 \cdot (n-1)^2 + 0.25 \cdot n \cdot (n-1). \end{aligned} \quad (2.132)$$

Подставим (2.132) и (2.127) в (2.129). Получим

$$\begin{aligned} \mathbf{D}\xi_n &= 0.25 \cdot n^2 \cdot (n-1)^2 + 0.25 \cdot n \cdot (n-1) - 0.25 \cdot n^2 \cdot (n-1)^2 = \\ &= 0.25 \cdot n \cdot (n-1). \end{aligned} \quad (2.133)$$

Применим к случайной величине ξ_n неравенство Чебышева

$$\mathbf{P}\{|\xi_n - \mathbf{M}\xi_n| \geq t\} \leq \frac{\mathbf{D}\xi_n}{t^2}. \quad (2.134)$$

Положим

$$\frac{\mathbf{D}\xi_n}{t^2} = \varepsilon. \quad (2.135)$$

По определению случайной величины ξ_n , левая часть неравенства (2.134) равна доле направленных графов, принадлежащих множеству $\overrightarrow{\mathbf{S}}_n^{dir}$, для которых число дуг m удовлетворяет неравенству

$$|m - 0.5 \cdot n \cdot (n - 1)| \geq t. \quad (2.136)$$

Следовательно, доля направленных графов, принадлежащих множеству $\overrightarrow{\mathbf{S}}_n^{dir}$, для которых число дуг m удовлетворяет неравенству

$$|m - 0.5 \cdot n \cdot (n - 1)| < t,$$

или, что эквивалентно, неравенствам

$$0.5 \cdot n \cdot (n - 1) - t < m < 0.5 \cdot n \cdot (n - 1) + t, \quad (2.137)$$

не меньше, чем

$$\mathbf{E}(\varepsilon) = 1 - \varepsilon. \quad (2.138)$$

Из равенства (2.135), с учетом равенства (2.133), находим

$$t = 0.5 \cdot \frac{1}{\sqrt{\varepsilon}} \cdot \sqrt{n \cdot (n - 1)}. \quad (2.139)$$

Подставим (2.139) в (2.137). Получаем, что доля направленных графов, принадлежащих множеству $\overrightarrow{\mathbf{S}}_n^{dir}$, для которых число дуг m удовлетворяет неравенствам

$$\begin{aligned} 0.5 \cdot n \cdot (n - 1) - 0.5 \cdot \frac{1}{\sqrt{\varepsilon}} \cdot \sqrt{n \cdot (n - 1)} < m < 0.5 \cdot n \cdot (n - 1) + \\ + 0.5 \cdot \frac{1}{\sqrt{\varepsilon}} \cdot \sqrt{n \cdot (n - 1)} \end{aligned} \quad (2.140)$$

не меньше, чем

$$\mathbf{E}(\varepsilon) = 1 - \varepsilon. \quad (2.141)$$

Положим

$$\varepsilon = \Theta(n^{-2\alpha}) \quad (n \rightarrow \infty), \quad (2.142)$$

где

$$\alpha \in (0;1). \quad (2.143)$$

Из (2.142) вытекает, что

$$0.5 \cdot \frac{1}{\sqrt{\varepsilon}} \cdot \sqrt{n \cdot (n-1)} = \Theta(n^{1+\alpha}) \quad (n \rightarrow \infty), \quad (2.144)$$

а из (2.141), (2.142) и (2.143) – что

$$E(\varepsilon) \rightarrow 1 \quad (n \rightarrow \infty). \quad (2.145)$$

Следовательно, из (2.140), (2.144) и (2.145) вытекает, что число m дуг почти всех направленных графов, принадлежащих множеству $\overrightarrow{\mathbf{S}}_n^{dir}$, удовлетворяет асимптотическому равенству $m = \Theta(n^2)$ ($n \rightarrow \infty$), что и требовалось доказать.

Теорема доказана.

Из теорем 2.7-2.10 непосредственно вытекает справедливость следующих следствий (доказательство которых аналогично доказательству следствий 2.1-2.4).

Следствие 2.7. Пусть выбор элементов множества $\overrightarrow{\mathbf{S}} \in \{\overrightarrow{\mathbf{S}}_n^{or}, \overrightarrow{\mathbf{S}}_n^{dir}\}$ осуществляется случайным образом. Тогда, в среднем, время, затрачиваемое на выполнение любой из операций $\setminus, \cup, \cap, \oplus$, а также на проверку истинности любого из отношений \subset и $=$ над представлениями элементов множества $\overrightarrow{\mathbf{S}}$ списками смежности вершин, асимптотически равно соответствующему времени для представлений элементов множества $\overrightarrow{\mathbf{S}}$ матрицами смежности вершин.

Следствие 2.8. Пусть выбор элементов множества $\overrightarrow{\mathbf{S}} \in \{\overrightarrow{\mathbf{S}}_n^{or}, \overrightarrow{\mathbf{S}}_n^{dir}\}$ осуществляется случайным образом. Тогда, в среднем, время, затрачиваемое на выполнение любой из операций $\setminus, \cup, \cap, \oplus$, а также на проверку истинности любого из отношений \subset и $=$ над представлениями элементов множества $\overrightarrow{\mathbf{S}}$ списками смежности дуг, асимптотически меньше, чем соответствующее время для представлений элементов множества $\overrightarrow{\mathbf{S}}$ матрицами смежности дуг.

Следствие 2.9. Для почти всех элементов множества $\overrightarrow{\mathbf{S}} \in \{\overrightarrow{\mathbf{S}}_n^{or}, \overrightarrow{\mathbf{S}}_n^{dir}\}$ ($n \rightarrow \infty$), время, затрачиваемое на выполнение любой из операций $\setminus, \cup, \cap, \oplus$, а также на проверку истинности любого из отношений \subset и $=$ над представлениями элементов множества $\overrightarrow{\mathbf{S}}$ списками смежности вершин, асимптотически равно соответствующему времени для

представлений представлениями элементов множества $\overline{\mathbf{S}}$ матрицами смежности вершин.

Следствие 2.10. Для почти всех элементов множества $\overline{\mathbf{S}} \in \{\overline{\mathbf{S}}_n^{or}, \overline{\mathbf{S}}_n^{dir}\} (n \rightarrow \infty)$, время, затрачиваемое на выполнение любой из операций $\setminus, \cup, \cap, \oplus$, а также на проверку истинности любого из отношений \subset и $=$ над представлениями элементов множества $\overline{\mathbf{S}}$ списками смежности дуг, асимптотически меньше, чем соответствующее время для представлений элементов множества $\overline{\mathbf{S}}$ матрицами смежности дуг.

Из теорем 2.7-2.10 также вытекает, что среднее число вершин и дуг у выбираемого случайным образом $G \in \mathbf{X}$, где $\mathbf{X} \in \{\bigcup_{k=1}^n \overline{\mathbf{S}}_n^{or}, \overline{\mathbf{T}}_n\}$, равно, соответственно, $\Theta(n)$ ($n \rightarrow \infty$) и $\Theta(n^2)$ ($n \rightarrow \infty$). Отсюда, в частности, вытекает, что среднее время выполнения соответствующим образом перенесенных на направленные графы операций $\circ, \rightarrow, \bullet, *, \wedge, \vee$ над случайно выбираемыми из множества $\mathbf{X} \in \{\bigcup_{k=1}^n \overline{\mathbf{S}}_n^{or}, \overline{\mathbf{T}}_n\}$ элементами удовлетворяет асимптотическим равенствам (2.76)-(2.81).

Кроме того, нетрудно показать, что теорема 2.6 остается истинной при переходе к множеству $\mathbf{X} \in \{\bigcup_{k=1}^n \overline{\mathbf{S}}_n^{or}, \overline{\mathbf{T}}_n\}$.

2.4. Выводы.

Проведенный в настоящем разделе анализ представлений графов и направленных графов дал возможность выявить ряд принципиальных моментов, связанных со сложностью этих представлений.

Первый момент связан с классом выполняемых над графами и/или направленными графами операций. Обычные теоретико-множественные операции не выводят за пределы класса $LCl(\mathbf{R})$. Однако, ситуация в корне изменяется, если допускаются такие операции, как имеющие многочисленные применения при разработке алгоритмов операции

$$\bullet, *, \wedge, \vee$$

(или их аналоги для направленных графов). В этом случае происходит, автоматический переход к классу $WORK - LCl(\mathbf{R})$. Это обстоятельство, в частности, подчеркивает значение класса $WORK - LCl(\mathbf{R})$ при разработке и анализе алгоритмов на графах (и/или направленных графах).

Второй момент связан с тем, что для всех рассматриваемых представлений класса $LCl(\mathbf{R})$ заведомо принадлежат задачи, решение которых сводится к последовательности проверок изоморфного включения в исходный граф (соответственно, направлен-

ный граф) последовательности графов (соответственно, направленных графов), генерируемой в рамках класса $LCI(\mathbf{R})$.

Третий момент связан с повсеместно применяемым представлением графов и направленных графов списками смежности вершин в прикладных исследованиях, связанных с разработкой алгоритмов. Интуитивно ясно, что такое представление оправдано для *разряженных* графов и/или направленных графов (каковыми, например, являются деревья). Однако, такое представление не оправданно, если действия (даже на уровне обычных теоретико-множественных операций) осуществляются на множестве всех графов (или орграфов, или направленных графов), имеющих фиксированное множество вершин. Действительно, как показано в настоящем разделе, в этом случае представление списками смежности вершин не дает выигрыша, по сравнению с представлениями матрицами смежности вершин, ни *в среднем*, ни *почти во всех случаях*.

Четвертый момент связан с представлениями списками и матрицами смежности ребер (для обычных графов) и дуг (для направленных графов). Эти представления, по своей сути, означают, для обычного графа переход к реберному графу, а для направленного графа – к реберному графу, вершины которого окрашены в четыре цвета. Реберный граф является конструкцией, более сложной, чем исходный объект. Однако преобразование

$$G \rightarrow L(G)$$

обладает следующим свойством. Пусть

$$\mathbf{Y} \in \{\mathbf{S}_n, \mathbf{T}_n, \bigcup_{k=1}^n \overline{\mathbf{S}_n}^{or}, \overline{\mathbf{T}_n}\}.$$

Для исходного множества объектов \mathbf{Y} представление списками смежности вершин не дает выигрыша, по сравнению с представлениями матрицами смежности вершин, ни *в среднем*, ни *почти во всех случаях*. Однако для образа $L(\mathbf{Y})$ исходного множества объектов \mathbf{Y} представление списками смежности вершин более предпочтительно, чем представление матрицами смежности вершин, как *в среднем*, так и *почти во всех случаях*.

3. КЛАССЫ $Lcl(\mathbf{R})$ И $WORK - Lcl(\mathbf{R})$.

Цель настоящего раздела состоит в характеристике основных задач теории графов с позиции их принадлежности классам $Lcl(\mathbf{R})$ и/или $WORK - Lcl(\mathbf{R})$. Одним из существенных обстоятельств, в силу которого значительное число задач теории графов принадлежит классу $WORK - Lcl(\mathbf{R})$, состоит в следующем: число конструируемых объектов – экспонента от числа вершин исходного графа, при условии, что задача построения одного (безразлично, какого именно) конструируемого объекта принадлежит классу $Lcl(\mathbf{R})$. Именно это обстоятельство и является одной из основных причин того, что настоящем разделе, наряду с разработкой алгоритмов решения конкретных задач, значительное внимание уделено построению нижних оценок числа конструируемых объектов.

Структура настоящего раздела – следующая. В п.3.1 сделан ряд замечаний, связанных с техникой разработки алгоритмов решения задач на графах. В п.3.2 детально исследуются задачи построения одного (безразлично, какого именно) и всех путей всех основных типов (кратчайших, неприводимых, гамильтоновых и т.д.) между двумя фиксированными вершинами графа. П.3.3 посвящен задачам построения одного (безразлично, какого именно) и всех циклов всех основных типов (кратчайших, неприводимых, и т.д.) проходящих через фиксированную вершину графа, а также задачи построения гамильтоновых циклов. В п.3.4 исследуются задачи построения остовных деревьев в связном графе, в том числе деревьев кратчайших путей и экономичных остовных деревьев (т.е. остовных деревьев наименьшей *стоимости*). В п.3.5 рассматривается *Rebbling* – специальный вид обхода ациклических орграфов, использующий разметку посещаемых вершин *камнями*. П.3.6 содержит краткие выводы.

3.1. Предварительные замечания.

Полученные в предыдущих разделах результаты показывают, что классам $Lcl(\mathbf{R})$ и $WORK - Lcl(\mathbf{R})$ принадлежит достаточно широкий класс алгоритмов на графах. В разделе 2 установлен ряд характеристик этих классов в терминах *теории алгебраических систем*. Наряду с методами теории алгебраических систем для решения дискретных задач широко применяется *поиск*, который, как известно, является, универсальным, а, часто, и единственным известным, методом их решения.

Рассмотрим общую характеристику классов $Lcl(\mathbf{R})$ и $WORK - Lcl(\mathbf{R})$ в терминах поиска.

Поиск может быть охарактеризован (см., напр., [9]) как метод построения такой конечной последовательности множеств объектов

$$Obj_0, Obj_1, \dots, Obj_l, \quad (3.1)$$

что:

- 1) множество Obj_0 представляет все допустимые объекты;
- 2) множество Obj_l представляет все решения;
- 3) включение

$$Obj_{i+1} \subset Obj_i$$

истинно для всех $i = 1, \dots, l-1$;

4) переход от множества Obj_i ($i = 0, 1, \dots, l$) к множеству Obj_{i+1} осуществляется методом решета, реализованного за счет применения специальным образом сформулированных *правил отсечения*.

Отметим, что:

- 1) число l , как правило, заранее не известно и определяется только по завершении построения последовательности (3.1);
- 2) множества

$$Obj_0, Obj_1, \dots, Obj_{l-1},$$

как правило, заданы в неявном виде, а множество Obj_l – в явном виде. При этом часто, но не всегда, множества

$$Obj_0, Obj_1, \dots, Obj_l$$

представляются с помощью дерева поиска.

В терминах поиска классы $Lcl(\mathbf{R})$ и $WORK - Lcl(\mathbf{R})$ характеризуются следующим образом.

Классу $Lcl(\mathbf{R})$ заведомо принадлежат следующие задачи на графах:

1. Все такие задачи на графах, что выполнены следующие два условия:

- а) множество Obj_0 всех допустимых объектов представимо в явном виде в рамках памяти $V(\mathbf{R}, n, m)$;
- б) предикат P проверки для объекта свойства «*быть решением*» вычислим в рамках памяти $V(\mathbf{R}, n, m)$.

2. Все такие задачи на графах, что выполнены следующие четыре условия:

- а) мощность множества Obj_0 всех допустимых объектов – экспонента от числа вершин исходных графов;
- б) существует алгоритм A , последовательно (один за другим) порождающий в явном виде элементы множества Obj_0 в рамках памяти $V(\mathbf{R}, n, m)$ (отсюда непосредственно вытекает, что каждый объект представим в явном виде в рамках памяти $V(\mathbf{R}, n, m)$);
- в) предикат P проверки для объекта свойства «*быть решением*» вычислим в рамках памяти $V(\mathbf{R}, n, m)$;
- г) множество решений представимо в явном виде в рамках памяти $V(\mathbf{R}, n, m)$.

Аналогичным образом, классу $WORK - Lcl(\mathbf{R}) \setminus Lcl(\mathbf{R})$ заведомо принадлежат все такие задачи на графах, что выполнены следующие четыре условия:

- а) мощность множества Obj_0 всех допустимых объектов – экспонента от числа вершин исходных графов;
- б) существует алгоритм A , последовательно (один за другим) порождающий в явном виде элементы множества Obj_0 в рамках памяти $V(\mathbf{R}, n, m)$;
- в) предикат P проверки для объекта свойства «*быть решением*» вычислим в рамках памяти $V(\mathbf{R}, n, m)$;
- г) мощность множества решений – экспонента от числа вершин исходных графов.

Замечание 3.1. Важная характеристика выделенного выше класса задач на графах вытекает из следующей характеристики задач поиска, полученной в [10] в рамках теоретико-множественного представления задач поиска (неявно заданным) *источником*

$$\mathbf{S} = (S, F, s_{in}, S_{fin}),$$

где S – конечное множество *ситуаций*, F – конечное множество (возможно, частичных) отображений множества S в себя, s_{in} ($s_{in} \in S$) – начальная ситуация, а S_{fin} ($\emptyset \neq S_{fin} \subset S$) – *множество финальных ситуаций*.

В [10] показано, что если F – порождающее множество коммутативной полугруппы, то емкостная сложность *поиска с возвратом (Backtracking)* равна

$$V = O(|F|) \quad (l \rightarrow \infty).$$

Следовательно, классам $Lcl(\mathbf{R})$ и $WORK - Lcl(\mathbf{R})$ заведомо принадлежат задачи на графах, решение которых осуществимо поиском с возвратом в рамках такого источника, что F – порождающее множество коммутативной полугруппы и применение каждого $f \in F$ осуществимо в рамках памяти $V(\mathbf{R}, n, m)$.

Применение поиска часто вызывает необходимость раскраски вершин и/или ребер исходного графа $G = (V, E) \in \mathbf{G}(n, m)$. Без ограничения общности можно считать, что раскраска вершин и/или ребер исходного графа осуществляется элементами множества \mathbf{Z}_q^k ($k \in \mathbf{N}$), где $\mathbf{Z}_q = \{0, 1, \dots, q-1\}$ (где q – фиксированное число). Объем памяти, необходимой для хранения раскраски вершин графа $G = (V, E)$ элементами множества \mathbf{Z}_q^k , равен $O(k \cdot |V|)$ ($|V| \rightarrow \infty$), если $k \rightarrow \infty$ и $O(|V|)$ ($|V| \rightarrow \infty$), если k – фиксированное число.

Аналогичным образом, объем памяти, необходимой для хранения раскраски ребер графа $G = (V, E)$ элементами множества \mathbf{Z}_q^k равен $O(k \cdot |E|)$ ($|E| \rightarrow \infty$), если $k \rightarrow \infty$ и $O(|E|)$ ($|E| \rightarrow \infty$), если k – фиксированное число.

Следовательно, использование раскрасок вершин и/или ребер исходного графа, восстанавливаемых в терминах задач, принадлежащих классам $Lcl(\mathbf{R})$ и $WORK - Lcl(\mathbf{R})$, не выводит исходную задачу за пределы соответствующих классов.

Замечание 3.2. Очевидно, что указанное выше свойство раскрасок вершин и/или ребер справедливо и для орграфов, и для направленных графов.

3.2. Задачи построения путей в графе.

Условимся $(v' - v'')$ -путь ($v' \neq v''$)

$$v' \{v', v_1\} v_1 \{v_1, v_2\} \dots v_{l-1} \{v_{l-1}, v''\} v'' \quad (3.2)$$

в графе $G = (V, E)$ записывать в виде

$$\pi_{v'-v''} = [v', v_1, \dots, v_{l-1}, v'']. \quad (3.3)$$

Замечание 3.3. Ясно, что записи (3.2) и (3.3) эквивалентны, т.е. по одной из них другая восстанавливается однозначно. Следовательно, принятое соглашение упрощает обозначения и не вызывает недоразумений.

Ясно, что задача построения множества всех $(v' - v'')$ -путей ($v' \neq v''$) в графе $G = (V, E) \in \mathbf{G}(n, m)$ заведомо принадлежит классу $WORK - Lcl(\mathbf{R})$.

Действительно, во-первых, каждый $(v' - v'')$ -путь представим в явном виде в рамках памяти $V(\mathbf{R}, n, m)$.

Во-вторых, проверка для последовательности

$$\pi_{v'-v''} = [v', v_1, \dots, v_{l-1}, v'']$$

вершин графа $G = (V, E)$ выполнения свойства: « $\pi_{v'-v''}$ – путь в графе $G = (V, E)$ » осуществима в рамках памяти $V(\mathbf{R}, n, m)$.

В-третьих, последовательная генерация (одна за другой) всех последовательностей

$$\pi_{v'-v''} = [v', v_1, \dots, v_{l-1}, v''] \quad (l = 1, \dots, |V| - 2)$$

попарно различных элементов множества V сводится к последовательной генерации (одного за другим) всех $(l - 1)$ -размещений ($l = 1, \dots, |V| - 2$) без повторов

$$(v_1, \dots, v_{i-1})$$

элементов множества $V \setminus \{v', v''\}$ (что, очевидно, осуществимо в рамках памяти $V(\mathbf{R}, n, m)$).

Рассмотрим задачи построения *кратчайших* $(v' - v'')$ -путей ($v' \neq v''$) в графе $G = (V, E) \in \mathbf{G}(n, m)$. Эти задачи являются модельными задачами теории графов и имеют многочисленные применения, как в самой теории графов, так и в ее приложениях.

Рассмотрим вначале задачу построения одного (безразлично, какого именно) кратчайшего $(v' - v'')$ -пути ($v' \neq v''$) в графе $G = (V, E) \in \mathbf{G}(n, m)$.

Традиционный подход к построению кратчайшего $(v' - v'')$ -пути ($v' \neq v''$) (см., напр., [2]) осуществляется в соответствии со следующим алгоритмом

Алгоритм 3.1.

Шаг 1. $U_0 := \{v'\}$, $i := 1$.

Шаг 2.

$$U_i := (\Gamma(U_{i-1}) \setminus (\bigcup_{j=0}^{i-1} U_j)) \setminus \{v \in \Gamma(U_{i-1}) \mid (v \neq v'') \& \\ \& (\Gamma(v) \setminus (\bigcup_{j=0}^{i-1} U_j) = \emptyset)\}.$$

Шаг 3. Если $U_i = \emptyset$, то переход к шагу 4, иначе – к шагу 5.

Шаг 4. $(v' - v'')$ -путь не существует и *конец*.

Шаг 5. Если $v'' \notin U_i$, то $i := i + 1$ и переход к шагу 2, иначе – к шагу 6.

Шаг 6. $v_i := v''$.

Шаг 7. Выбрать вершину $v \in \Gamma(v_i) \cap U_{i-1}$, $i := i - 1$ и $v_i := v$.

Шаг 8. Если $i \neq 0$, то переход к шагу 7, иначе

$$\pi_{v'-v''} := [v_0, v_1, \dots, v_{i-1}, v_i]$$

и *конец*.

Корректность алгоритма 3.1 непосредственно вытекает из того, что

$$U_i \subseteq \{v \in V \mid d(v', v) = i\} \quad (i = 0, 1, \dots)$$

и ни одна из вершин, принадлежащих множеству

$$\{v \in V \mid d(v', v) = i\} \setminus U_i \quad (i = 0, 1, \dots)$$

не принадлежит никакому кратчайшему $(v' - v'')$ -пути.

Замечание 3.4. Алгоритм 3.1 легко переносится на случай направленных графов. Действительно, Шаги 1, 3-6, 8 остаются без изменения, Шаг 2 заменяется на

Шаг 2'.

$$U_i := (\Gamma^+(U_{i-1}) \setminus (\bigcup_{j=0}^{i-1} U_j)) \setminus \{v \in \Gamma^+(U_{i-1}) \mid (v \neq v'') \& \\ \& (\Gamma^+(v) \setminus (\bigcup_{j=0}^{i-1} U_j) = \emptyset)\},$$

а Шаг 7 – на

Шаг 7'. Выбрать вершину $v \in \Gamma^-(v_i) \cap U_{i-1}$, $i := i - 1$ и $v_i := v$.

Учитывая, что указанный выше переход от графов к направленным графам не вызывает сложности, в дальнейшем, чтобы не загромождать изложение излишними деталями там, где это не вызывает необходимость, ограничимся рассмотрением обычных графов.

Из алгоритма 3.1 вытекает, что задача построения одного (безразлично, какого именно) кратчайшего $(v' - v'')$ -пути $(v' \neq v'')$ в графе $G = (V, E) \in \mathbf{G}(n, m)$ принадлежит классу $LC/(\mathbf{R})$, где \mathbf{R} – представление графов, либо списками, либо матрицами смежности (либо вершин, либо ребер).

Действительно, алгоритм 3.1 при каждом $i = 0, 1, \dots$ состоит, по своей сути, в преобразовании матрицы \mathbf{M} порядка $|V| \times 4$, у которой 1-й столбец – характеристическая функция множества $\bigcup_{j=0}^{i-1} U_j$, 2-й столбец – характеристическая функция множества U_i , 3-й столбец – характеристическая функция следующего восстанавливаемого множества U_{i+1} , а 4-й столбец указывает позицию соответствующей вершины в восстанавливаемом $(v' - v'')$ -пути $(v' \neq v'')$, что непосредственно вытекает из истинности равенств

$$U_{i+1} := (\Gamma(U_i) \setminus ((\bigcup_{j=0}^{i-1} U_j) \cup U_i)) \setminus \{v \in \Gamma(U_i) \mid (v \neq v'') \& \\ \& (\Gamma(v) \setminus ((\bigcup_{j=0}^{i-1} U_j) \cup U_i) = \emptyset)\}. \quad (3.4)$$

$$\bigcup_{j=0}^{i-1} U_j = (\bigcup_{j=0}^i U_j) \setminus U_i, \quad (3.5)$$

$$U_{i-1} = \left(\bigcup_{j=0}^{i-1} U_j \right) \cap \Gamma(U_i). \quad (3.6)$$

Существенно, что преобразования (3.4)-(3.6) над матрицей \mathbf{M} осуществимы с емкостной сложностью $O(|V|)$ ($|V| \rightarrow \infty$) при равномерном весе (и с емкостной сложностью $O(|V| \cdot \log |V|)$ ($|V| \rightarrow \infty$) при логарифмическом весе).

Замечание 3.5. Отмеченная выше характеристика означает, что, по своей сути, алгоритм 3.1 сводится к преобразованиям некоторой раскраски вершин исходного графа $G = (V, E)$ элементами множества $\mathbf{Z}_2^3 \times \mathbf{Z}_{|V|}$, осуществляемой с емкостной сложностью $O(|V|)$ ($|V| \rightarrow \infty$) при равномерном весе (и с емкостной сложностью $O(|V| \cdot \log |V|)$ ($|V| \rightarrow \infty$) при логарифмическом весе).

Очевидно, что подграф графа $G = (V, E)$, индуцированный множеством вершин $\bigcup_{j=0}^i U_j$, построенным к моменту выхода из цикла, определяемого шагами 1-5 алгоритма 3.1, представляет в неявном виде множество всех кратчайших $(v' - v'')$ -путей ($v' \neq v''$) в графе $G = (V, E)$. Небольшая модификация алгоритма 3.1 дает возможность преобразовать его в алгоритм построения в явном виде множества всех кратчайших $(v' - v'')$ -путей ($v' \neq v''$) в графе $G = (V, E) \in \mathbf{G}(n, m)$.

Действительно, чтобы построить множество всех кратчайших $(v' - v'')$ -путей ($v' \neq v''$) в графе $G = (V, E)$, достаточно реорганизовать шаги 5-8 алгоритма 3.1 так, чтобы был реализован поиск с возвратом (*Backtracking*) по системе множеств

$$U_{i-1}, \dots, U_1.$$

При этом каждый раз достаточно хранить только очередной восстанавливаемый кратчайший путь, так что рабочая память алгоритма построения множества всех кратчайших $(v' - v'')$ -путей ($v' \neq v''$) в графе $G = (V, E)$ асимптотически равна рабочей памяти алгоритма 3.1.

Таким образом, задача построения в явном виде множества всех кратчайших $(v' - v'')$ -путей ($v' \neq v''$) в графе $G = (V, E) \in \mathbf{G}(n, m)$ принадлежит классу $WORK - LCl(\mathbf{R})$, где \mathbf{R} – представление графов, либо списками, либо матрицами смежности (либо вершин, либо ребер). Из следующей теоремы вытекает, что эта задача не принадлежит классу $LCl(\mathbf{R})$ при любом представлении \mathbf{R} графа $G = (V, E) \in \mathbf{G}(n, m)$.

Теорема 3.1. Пусть $L_n^{kp}(G)$ ($n \in \mathbf{N}$) – максимальное число кратчайших путей между двумя фиксированными вершинами в графе $G \in \mathbf{S}_n$ и

$$L_n^{kp} = \max_{G \in \mathbf{S}_n} L_n^{kp}(G).$$

Если $n \equiv 2 \pmod{3}$, то

$$L_n^{kp} \geq (\sqrt[3]{3})^{n-2}. \quad (3.7)$$

Доказательство. Очевидно, что теорема справедлива, если $n \in \{2,5\}$. Действительно,

$$L_2^{kp} = 1,$$

$$L_5^{kp} = 3,$$

так что неравенство (3.7) выполнено, если $n \in \{2,5\}$.

Пусть $n \equiv 2 \pmod{3}$ и $n > 5$. Выберем из множества

$$\mathbf{G}(n, k_1 + k_1 \cdot k_2 + k_2 \cdot k_3 + \dots + k_{l-2} \cdot k_{l-1} + k_{l-1}) \quad (n, l \in \mathbf{N}, l \geq 3),$$

где

$$k_1 + k_2 + \dots + k_{l-1} + 2 = n \quad (k_1, k_2, \dots, k_{l-1} \in \mathbf{N}), \quad (3.8)$$

такой граф $G_1(k_1, \dots, k_{l-1}) = (V, E_1)$, что

$$\begin{aligned} E_1 = & \{ \{1, i\} \mid i = 2, \dots, k_1 + 1 \} \cup \\ & \cup \{ \{1 + k_1 + \dots + k_{l-2} + i, n\} \mid i = 1, \dots, k_{l-1} \} \cup \\ & \cup \left(\bigcup_{i=0}^{l-3} \{ \{1 + \sum_{j=1}^i k_j + u, 1 + \sum_{j=1}^{i+1} k_j + v\} \mid u = 1, \dots, k_{i+1}; v = 1, \dots, k_{i+2} \} \right). \end{aligned}$$

Число кратчайших $(1, n)$ -путей в графе $G_1(k_1, \dots, k_{l-1})$ равно

$$f(k_1, \dots, k_{l-1}) = \prod_{i=1}^{l-1} k_i. \quad (3.9)$$

Известно, что функция (3.9), при выполнении условия

$$k_1 + k_2 + \dots + k_{l-1} + 2 = n \quad (k_1, k_2, \dots, k_{l-1} > 0),$$

достигает своего наибольшего значения, если

$$k_1 = k_2 = \dots = k_{l-1} = k. \quad (3.10)$$

Это наибольшее значение функции f равно

$$f(k, \dots, k) = k^{l-1}. \quad (3.11)$$

Из равенства

$$(l-1) \cdot k + 2 = n$$

находим

$$l-1 = \frac{n-2}{k}. \quad (3.12)$$

Подставим (3.12) в (3.11). Получим

$$f(k, \dots, k) = \left(k^{\frac{1}{k}}\right)^{n-2}.$$

Известно, что на множестве \mathbf{N} функция

$$g(k) = k^{\frac{1}{k}}$$

принимает наибольшее значение, если $k = 3$. Следовательно, если $n \equiv 2 \pmod{3}$, то число кратчайших $(1, n)$ -путей в графе $G_1(\underbrace{3, 3, \dots, 3}_{l-1 \text{ раз}})$ равно

$$f(\underbrace{3, 3, \dots, 3}_{l-1 \text{ раз}}) = (\sqrt[3]{3})^{n-2}. \quad (3.13)$$

Из (3.13) непосредственно вытекает справедливость неравенства (3.7), что и требовалось доказать.

Теорема доказана.

Замечание 3.6. Рассмотрим бесконечную последовательность значений

$$n = 3 \cdot k + 2 \quad (k \in \mathbf{N}).$$

Из доказательства теоремы 3.1, в частности, вытекает, что если $n \rightarrow \infty$, то задача построения в явном виде множества всех кратчайших $(v' - v'')$ -путей $(v' \neq v'')$ в графе $G \in \mathbf{G}(n, 3 \cdot n - 9)$ не принадлежит классу $LCI(\mathbf{R})$ при любом представлении \mathbf{R} графа G .

Метод, которым алгоритм 3.1 осуществляет построение кратчайшего $(v' - v'')$ -пути $(v' \neq v'')$ в графе $G = (V, E) \in \mathbf{G}(n, m)$, характеризуется как *восстановление начальных отрезков*.

Действительно, шаги 1-5 алгоритма 1.3 выделяют потенциально возможные начальные отрезки кратчайших $(v' - v'')$ -путей $(v' \neq v'')$, а затем из них шаги 6-8 осуществляют выбор конкретного кратчайшего $(v' - v'')$ -пути (см. рис. 3.1 а)).

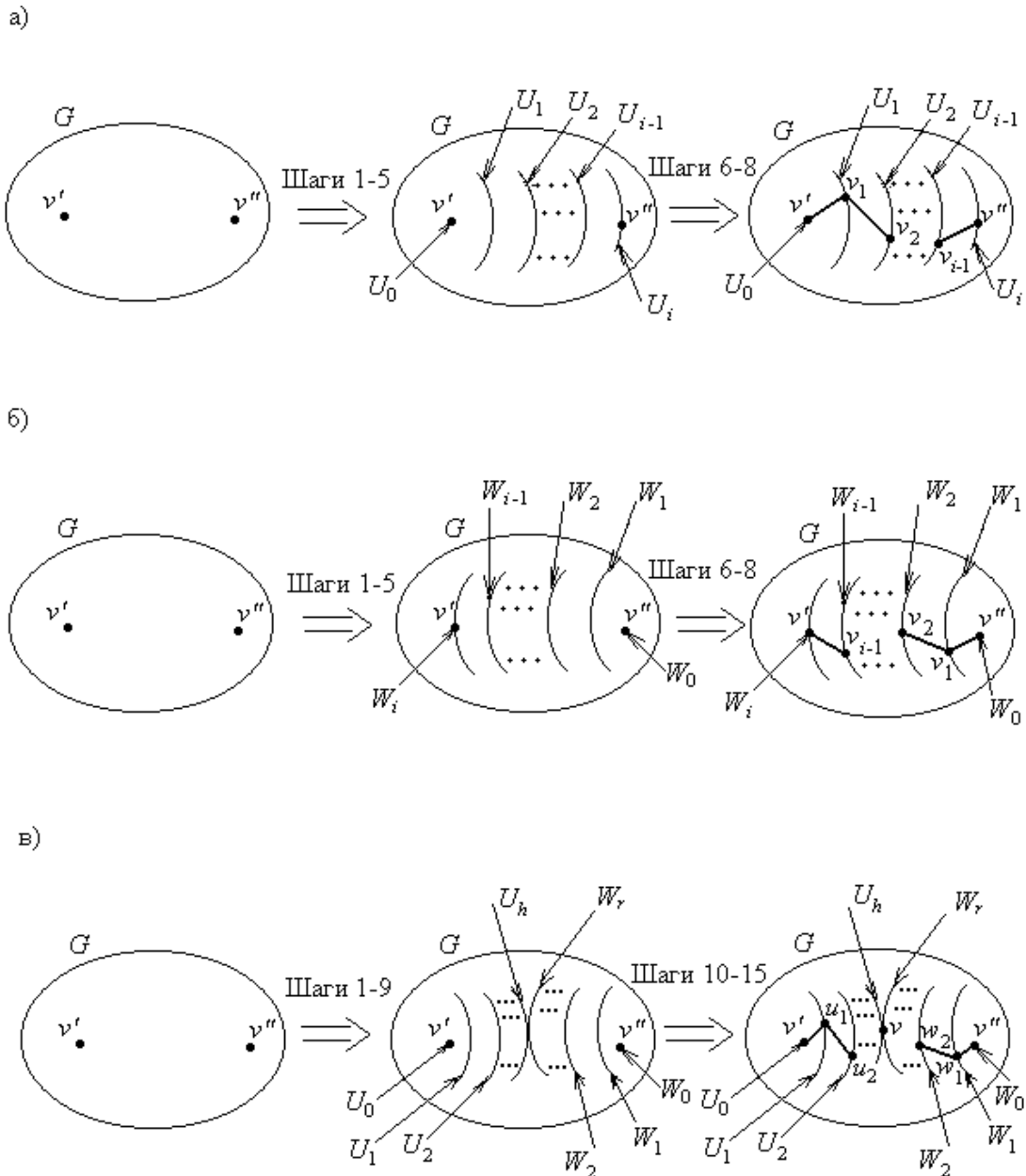


Рис. 3.1. Построение кратчайшего $(v' - v'')$ -пути в графе G : а) восстановлением начальных отрезков; б) восстановлением финальных отрезков; в) двусторонним поиском.

Рассмотрим, в некотором смысле, *двойственный* метод построения одного (безразлично, какого именно) кратчайшего $(v' - v'')$ -пути $(v' \neq v'')$ в графе $G = (V, E) \in \mathbf{G}(n, m)$, который может быть охарактеризован как *восстановление финальных отрезков*. Этот метод состоит в том, что вначале выделяют потенциально возможные финальные отрезки кратчайших $(v' - v'')$ -путей, а затем из них осуществляется выбор конкретного кратчайшего $(v' - v'')$ -пути (см. рис. 3.1 б)). Соответствующий алгоритм получается в результате небольшой модификации алгоритма 3.1, без увеличения его емкостной сложности, и имеет следующий вид

Алгоритм 3.2.

Шаг 1. $W_0 := \{v''\}$, $i := 1$.

Шаг 2.

$$W_i := (\Gamma(W_{i-1}) \setminus (\bigcup_{j=0}^{i-1} W_j)) \setminus \{v \in \Gamma(W_{i-1}) \mid (v \neq v') \& \\ \& (\Gamma(v) \setminus (\bigcup_{j=0}^{i-1} W_j) = \emptyset)\}.$$

Шаг 3. Если $W_i = \emptyset$, то переход к шагу 4, иначе – к шагу 5.

Шаг 4. $(v' - v'')$ -путь не существует и *конец*.

Шаг 5. Если $v' \notin W_i$, то $i := i + 1$ и переход к шагу 2, иначе – к шагу 6.

Шаг 6. $v_i := v'$.

Шаг 7. Выбрать вершину $v \in \Gamma(v_i) \cap W_{i-1}$, $i := i - 1$ и $v_i := v$.

Шаг 8. Если $i \neq 0$, то переход к шагу 7, иначе

$$\pi_{v'-v''} := [v_i, v_{i-1}, \dots, v_1, v_0]$$

и *конец*.

Алгоритмы 3.1 и 3.2 дают возможность без увеличения емкостной сложности осуществлять *двусторонний поиск* при решении задачи построения кратчайшего $(v' - v'')$ -пути $(v' \neq v'')$ в графе $G = (V, E) \in \mathbf{G}(n, m)$, т.е. *распараллелить вычисления* за счет одновременного восстановления как начальных, так и финальных отрезков (см. рис. 3.1.в)). Соответствующий алгоритм имеет следующий вид

Алгоритм 3.3.

Шаг 1. Если $\{v', v''\} \in E$, то

$$\pi_{v'-v''} := [v', v'']$$

и *конец*, иначе переход к шагу 2.

Шаг 2. $U_0 := \{v'\}$, $W_0 := \{v''\}$, $h := 0$, $r := 0$.

Шаг 3. Если $|U_h| < |W_r|$, то переход к шагу 4, иначе – к шагу 6.

Шаг 4. $h := h + 1$,

$$U_h := (\Gamma(U_{h-1}) \setminus (\bigcup_{j=0}^{h-1} U_j)) \setminus \{v \in \Gamma(U_{h-1}) \mid (v \neq v'') \& \\ \& (\Gamma(v) \setminus (\bigcup_{j=0}^{h-1} U_j) = \emptyset)\}.$$

Шаг 5. Если $U_h = \emptyset$, то переход к шагу 8, иначе – к шагу 9.

Шаг 6. $r := r + 1$,

$$W_r := (\Gamma(W_{r-1}) \setminus (\bigcup_{j=0}^{r-1} W_j)) \setminus \{v \in \Gamma(W_{r-1}) \mid (v \neq v') \& \\ \& (\Gamma(v) \setminus (\bigcup_{j=0}^{r-1} W_j) = \emptyset)\}.$$

Шаг 7. Если $W_r = \emptyset$, то переход к шагу 8, иначе – к шагу 9.

Шаг 8. $(v' - v'')$ -путь не существует и *конец*.

Шаг 9. Если $U_h \cap W_r \neq \emptyset$, то переход к шагу 10, иначе – к шагу 3.

Шаг 10. Выбрать вершину $v \in U_h \cap W_r$, $u_h := v$, $w_r := v$.

Шаг 11. Если $h \neq 0$, то переход к шагу 12, иначе – к шагу 13.

Шаг 12. Выбрать вершину $v \in \Gamma(u_h) \cap U_{h-1}$, $h := h - 1$, $u_h := v$ и переход к шагу 11.

Шаг 13. Если $r \neq 0$, то переход к шагу 14, иначе – к шагу 15.

Шаг 14. Выбрать вершину $v \in \Gamma(w_r) \cap W_{r-1}$, $r := r - 1$, $w_r := v$ и переход к шагу 11.

Шаг 15.

$$\pi_{v'-v''} := [u_0, u_1, \dots, u_h, w_{r-1}, \dots, w_1, w_0]$$

и *конец*.

Пример 3.1. В качестве приложения полученных выше результатов рассмотрим проверку свойства «быть связным» для графа $G = (V, E) \in \mathbf{S}_n$ (принадлежность этой задачи классу $LCI(\mathbf{R})$, фактически, была показана в примерах 1.7 и 1.10).

Решение этой задачи, существенно использующее структуру открытых окрестностей вершин графа G , состоит в следующем. Достаточно зафиксировать вершину $v \in V$ графа G и построить (с помощью любого из алгоритмов 3.1-3.3) кратчайший путь $\pi_{v-v''}$ в каждую вершину $v'' \in V \setminus \{v\}$ (при этом, очевидно, что для вершин, лежащих на построенных ранее путях строить пути не нужно). Если указанные пути существуют для каждой вершины $v'' \in V \setminus \{v\}$, то G – связный граф. Если же хотя бы для одной вершины $v'' \in V \setminus \{v\}$ указанный путь не существует, то граф G не является связным.

Важным свойством того или иного объекта, имеющим многочисленные приложения, является свойство «*быть не избыточным*». Применительно к путям в графе это свойство может быть формализовано следующим образом.

Определение 3.1. $(v' - v'')$ -путь $(v' \neq v'')$

$$\pi_{v'-v''} = [v_0, v_1, \dots, v_{l-1}, v_l] \quad (v_0 = v', v_l = v'')$$

в графе $G = (V, E)$ назовем *неприводимым*, если

$$(\forall i, j \in \{0, 1, \dots, l\})(|i - j| \geq 2 \Rightarrow \{v_i, v_j\} \notin E). \quad (3.14)$$

Очевидно, что каждый кратчайший путь всегда является неприводимым путем. Это означает, что справедлива следующая оценка.

Следствие 3.1. Пусть $L_n^{непр}(G)$ ($n \in \mathbf{N}$) – максимальное число неприводимых путей между двумя фиксированными вершинами в графе $G \in \mathbf{S}_n$ и

$$L_n^{непр} = \max_{G \in \mathbf{S}_n} L_n^{непр}(G).$$

Если $n \equiv 2 \pmod{3}$, то

$$L_n^{непр} \geq (\sqrt[3]{3})^{n-2}. \quad (3.15)$$

Доказательство. В графе $G_1(\underbrace{3, 3, \dots, 3}_{l-1 \text{ раз}})$ из теоремы 3.1 множества кратчайших и неприводимых $(1, n)$ -путей совпадают. Следовательно, справедливость оценки (3.15) непосредственно вытекает из оценки (3.7).

Следствие доказано.

Замечание 3.7. Рассмотрим бесконечную последовательность значений

$$n = 3 \cdot k + 2 \quad (k \in \mathbf{N}).$$

Из доказательства следствия 3.1, в частности, вытекает, что если $n \rightarrow \infty$, то задача построения в явном виде множества всех неприводимых $(v' - v'')$ -путей $(v' \neq v'')$ в графе

$$G \in \mathbf{G}(n, 3 \cdot n - 9)$$

не принадлежит классу $LCl(\mathbf{R})$ при любом представлении \mathbf{R} графа G .

Основное значение следствия 3.1 состоит в том, что из него вытекает, что существует такой закон роста $m \rightarrow \infty$, что задача построения в явном виде множества всех неприводимых $(v' - v'')$ -путей $(v' \neq v'')$ в графе $G \in \mathbf{G}(n, m)$ не принадлежит классу $LCl(\mathbf{R})$ при любом представлении \mathbf{R} графа G . Тем не менее, при любом (допустимом) законе роста $m \rightarrow \infty$ эта задача принадлежит классу $WORK - LCl(\mathbf{R})$, где \mathbf{R} – представление графа $G = (V, E) \in \mathbf{G}(n, m)$, либо списками, либо матрицами смежности (либо вершин, либо ребер), в чем легко убедиться, организовав поиск с возвращением (*Backtracking*), каждый раз сохраняя только очередной восстанавливаемый неприводимый путь. Соответствующий алгоритм имеет следующий вид.

Алгоритм 3.4.

Шаг 1. $v_0 := v'$, $U_0 := \emptyset$, $i := 0$, $\pi := [v_0, \dots, v_{i-1}]$.

/ прямой ход */*

Шаг 2. $i := i + 1$, $U_i := \Gamma(U_{i-1})$.

Шаг 3. Если $U_i \neq \emptyset$, то переход к шагу 4, иначе – к шагу 8.

Шаг 4. Если существует такое $v \in U_i$, что $\{v, v_j\} \notin E$ для всех $j \in \{0, 1, \dots, i-2\}$, то переход к шагу 5, иначе – к шагу 8.

Шаг 5. $v_i := v$, $\pi := [v_0, \dots, v_{i-1}, v_i]$, $U_i := U_i \setminus \{v\}$.

Шаг 6. Если $v_i \neq v''$, то переход к шагу 2, иначе – к шагу 7.

Шаг 7. Выдать $\pi := [v_0, \dots, v_{i-1}, v_i]$, $\pi := [v_0, \dots, v_{i-1}]$.

/ обратный ход */*

Шаг 8. $i := i - 1$.

Шаг 9. Если $i \neq 0$, то $\pi := [v_0, \dots, v_{i-1}]$ и переход к шагу 3, иначе *конец*.

Рассмотрим теперь задачи построения *гамильтоновых* $(v' - v'')$ -путей $(v' \neq v'')$ в заданном графе $G = (V, E) \in \mathbf{G}(n, m)$. Эти задачи являются модельными задачами теории графов и имеют многочисленные применения, как в самой теории графов, так и в ее приложениях.

Задача построения в явном виде одного (безразлично, какого именно) гамильтонового $(v' - v'')$ -пути $(v' \neq v'')$ в графе $G = (V, E) \in \mathbf{G}(n, m)$ принадлежит классу $LCl(\mathbf{R})$, где \mathbf{R} – представление графов, либо списками, либо матрицами смежности (либо вершин, либо ребер).

Действительно, достаточно последовательно генерировать (одно за другим) $(n - 2)$ -размещения без повторений элементов множества $V \setminus \{v', v''\}$, каждый раз сохраняя только текущее размещение (очевидно, что эта задача принадлежит классу $LCl(\mathbf{R})$).

Для каждой последовательности

$$\pi = [v', v_{j_1}, \dots, v_{j_{n-2}}, v'']$$

проверяется, выполнено ли свойство:

$$\langle \pi - \text{путь в графе } G = (V, E) \rangle.$$

При положительном ответе происходит остановка вычислений, так как искомый гамильтонов путь найден. Если осуществлена проверка всех указанных выше последовательностей и положительный ответ не найден, то искомый путь не существует.

Очевидно, что задача построения множества всех гамильтоновых $(v' - v'')$ -путей $(v' \neq v'')$ в графе $G = (V, E) \in \mathbf{S}_n$ не принадлежит классу $LCl(\mathbf{R})$ при любом представлении \mathbf{R} графа G .

Действительно, известно, что число гамильтоновых $(v' - v'')$ -путей $(v' \neq v'')$ в полном графе K_n равно $(n - 2)!$. Однако, отличительной особенностью графа K_n является то, что он – экстремальный граф (так как $K_n \in \mathbf{G}(n, \binom{n}{2})$).

Покажем, что экспоненциальная оценка для числа гамильтоновых путей, соединяющих две фиксированные вершины графа $G \in \mathbf{G}(n, m)$, достижима уже в случае, когда

$$m = \Theta(n) \quad (n \rightarrow \infty).$$

Теорема 3.2. Пусть $L_{n,m}^{зам}(G)$ ($n, m \in \mathbf{N}, 0 \leq m \leq \binom{n}{2}$) – максимальное число гамильтоновых путей между двумя фиксированными вершинами в графе $\mathbf{G}(n, m)$ и

$$L_{n,m}^{зам} = \max_{G \in \mathbf{G}(n,m)} L_{n,m}^{зам}(G).$$

Если $n = 3 \cdot k + 2$ ($k \in \mathbf{N}$), то

$$L_{n,4 \cdot n - 11}^{зам} \geq 0.5 \cdot (\sqrt[3]{12})^{n-2}. \quad (3.16)$$

Доказательство. Пусть $n \equiv 2 \pmod{3}$ и $n \geq 5$. Рассмотрим граф

$$G_2(\underbrace{3, 3, \dots, 3}_{\frac{n-2}{3} \text{ раз}}) = (V, E_2),$$

полученный в результате следующей небольшой модификации графа

$$G_1(\underbrace{3, 3, \dots, 3}_{\frac{n-2}{3} \text{ раз}}) = (V, E_1)$$

из теоремы 3.1:

$$E_2 = E_1 \cup \left(\bigcup_{i=0}^{\frac{n-2}{3}-1} \{ \{2+3 \cdot i, 3+3 \cdot i\}, \{2+3 \cdot i, 4+3 \cdot i\}, \{3+3 \cdot i, 4+3 \cdot i\} \} \right). \quad (3.17)$$

Из (3.17) вытекает, что

$$G_2(\underbrace{3, 3, \dots, 3}_{\frac{n-2}{3} \text{ раз}}) \in \mathbf{G}(n, 4 \cdot n - 11).$$

Будем говорить, что гамильтонов $(1, n)$ -путь в графе $G_2(\underbrace{3, 3, \dots, 3}_{\frac{n-2}{3} \text{ раз}})$

имеет тип

$$(i_1, i_2, \dots, i_l) \quad (i_1 + i_2 + \dots + i_l = \frac{n-2}{3}; i_1 > 0, i_2 > 0, \dots, i_l > 0),$$

если этот путь содержит l «зигзагов», соответственно, «размеров» i_1, i_2, \dots, i_l .

Структура гамильтонового $(1, n)$ -пути типа (i_1, i_2, \dots, i_l) показана на рис. 3.2.

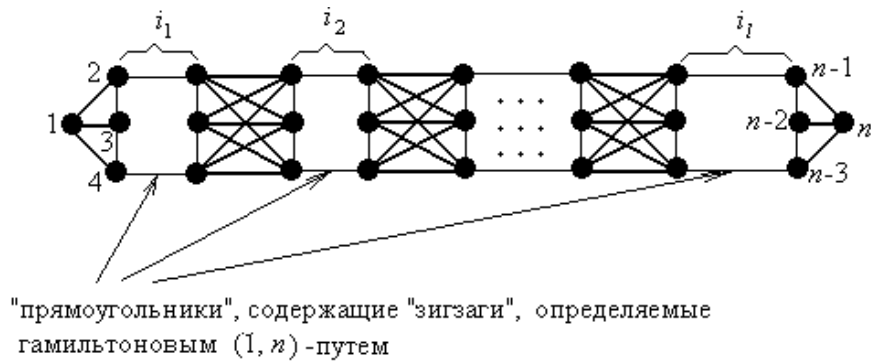


Рис. 3.2. Структура гамильтонового $(1, n)$ -пути типа (i_1, i_2, \dots, i_l)

$$(i_1 + i_2 + \dots + i_l = \frac{n-2}{3}; i_1 > 0, i_2 > 0, \dots, i_l > 0) \text{ в графе } G_2(3, 3, \dots, 3).$$

Типы возможных «зигзагов» изображены на рис. 3.3.

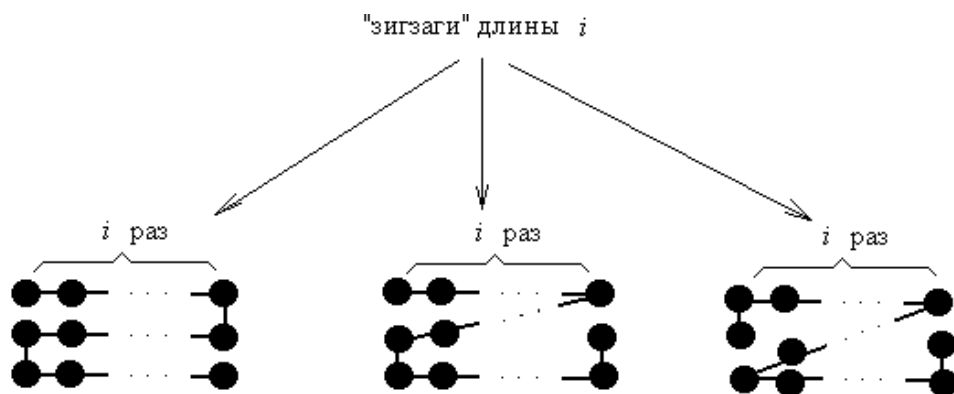


Рис. 3.3. Типы допустимых "зигзагов" в гамильтоновом $(1, n)$ -пути в графе $G_2(3, 3, \dots, 3)$.

Ясно, что общее число допустимых «зигзагов» «размера» i ($i = 1, \dots, \frac{n-2}{3}$) не меньше, чем 6^i . Следовательно, число гамильтоновых $(1, n)$ -путей типа

$$(i_1, i_2, \dots, i_l) \quad (i_1 + i_2 + \dots + i_l = \frac{n-2}{3}; i_1 > 0, i_2 > 0, \dots, i_l > 0)$$

в графе $G_2(\underbrace{3, 3, \dots, 3}_{\frac{n-2}{3} \text{ раз}})$ не меньше, чем

$$6^{i_1} \cdot 6^{i_2} \cdot \dots \cdot 6^{i_l} = 6^{i_1+i_2+\dots+i_l} = 6^{\frac{n-2}{3}}.$$

Отсюда вытекает, что

$$\begin{aligned} L_{n, 4 \cdot n-11}^{\text{гам}} &\geq \sum_{l=1}^{\frac{n-2}{3}} \sum_{\substack{(i_1, \dots, i_l) \\ i_1 > 0, \dots, i_l > 0 \\ i_1 + \dots + i_l = \frac{n-2}{3}}} 6^{\frac{n-2}{3}} = \sum_{l=1}^{\frac{n-2}{3}} 6^{\frac{n-2}{3}} \cdot \sum_{\substack{(i_1, \dots, i_l) \\ i_1 > 0, \dots, i_l > 0 \\ i_1 + \dots + i_l = \frac{n-2}{3}}} 1 = 6^{\frac{n-2}{3}} \cdot \sum_{l=1}^{\frac{n-2}{3}} \binom{\frac{n-2}{3} - 1}{l-1} = \\ &= 6^{\frac{n-2}{3}} \cdot \sum_{l=1}^{\frac{n-2}{3}-1} \binom{\frac{n-2}{3} - 1}{l-1} = 6^{\frac{n-2}{3}} \cdot 2^{\frac{n-2}{3}-1} = 0.5 \cdot (\sqrt[3]{12})^{n-2}, \end{aligned}$$

что и требовалось доказать.

Теорема доказана.

Из теоремы 3.2, в частности, вытекает, что существует такой закон роста $m \rightarrow \infty$, удовлетворяющий условию

$$m = \Theta(n) \quad (n \rightarrow \infty),$$

что задача построения в явном виде множества всех гамильтоновых $(v' - v'')$ -путей ($v' \neq v''$) в графе $G \in \mathbf{G}(n, m)$ не принадлежит классу $LCl(\mathbf{R})$ при любом представлении \mathbf{R} графа G . Тем не менее, эта задача всегда принадлежит классу $WORK-LCl(\mathbf{R})$, где \mathbf{R} – представление графов, либо списками, либо матрицами смежности (либо вершин, либо ребер).

Действительно, достаточно организовать поиск с возвратом (*Backtracking*), каждый раз сохраняя только очередной восстанавливаемый гамильтонов путь. Соответствующий алгоритм имеет следующий вид.

Алгоритм 3.5.

Шаг 1. $v' := v_0, U_0 := \emptyset, i := 0, \pi := [v_0, \dots, v_{i-1}]$.

/ прямой ход */*

Шаг 2. $i := i + 1, U_i := \Gamma(U_{i-1})$.

Шаг 3. Если $i < |V| - 1$, то $U_i := U_i \setminus \{v''\}$.

Шаг 4. Если $U_i \neq \emptyset$, то переход к шагу 4, иначе – к шагу 10.

Шаг 5. Если существует такое $v \in U_i$, что $v \neq v_j$ для всех $j \in \{0, 1, \dots, i-1\}$, то переход к шагу 5, иначе – к шагу 10.

Шаг 6. $v_i := v, \pi := [v_0, \dots, v_{i-1}, v_i], U_i := U_i \setminus \{v\}$.

Шаг 7. Если $i < |V| - 1$, то переход к шагу 2, иначе – к шагу 8.

Шаг 8. Выдать $\pi := [v_0, \dots, v_{i-1}, v_i], \pi := [v_0, \dots, v_{i-1}]$.

/ обратный ход */*

Шаг 9. $i := i - 1, \pi := [v_0, \dots, v_{i-1}]$ и переход к шагу 4.

Шаг 10. Если $i = 0$, то *конец*, иначе переход к шагу 12.

В некотором смысле двойственными к задаче построения в явном виде множества всех кратчайших $(v' - v'')$ -путей $(v' \neq v'')$ в графе $G = (V, E) \in \mathbf{G}(n, m)$ являются задачи построения в явном виде множеств, как всех самых длинных, так и всех самых длинных неприводимых $(v' - v'')$ -путей $(v' \neq v'')$ в графе $G = (V, E)$. Справедливы следующие оценки.

Теорема 3.3. Пусть $L_{n,m}^{\max}(G)$ и $L_{n,m}^{\max-непр}(G)$ ($n, m \in \mathbf{N}, 0 \leq m \leq \binom{n}{2}$) – максимальное число, соответственно, самых длинных и самых длинных неприводимых путей между двумя фиксированными вершинами в графе $G \in \mathbf{G}(n, m)$,

$$L_{n,m}^{\max} = \max_{G \in \mathbf{G}(n,m)} L_{n,m}^{\max}(G),$$

$$L_{n,m}^{\max-непр} = \max_{G \in \mathbf{G}(n,m)} L_{n,m}^{\max-непр}(G).$$

Если $n = 3 \cdot k + 2$ ($k \in \mathbf{N}$), то

$$L_{n,4n-11}^{\max} \geq 0.5 \cdot (\sqrt[3]{12})^{n-2}. \quad (3.18)$$

и

$$L_{n,3n-9}^{\max-непр} \geq (\sqrt[3]{3})^{n-2}. \quad (3.19)$$

Доказательство. В графе

$$G_2(\underbrace{3,3,\dots,3}_{\frac{n-2}{3} \text{ раз}}) \quad (n = 3 \cdot k + 2, (k \in \mathbf{N}))$$

из теоремы 3.2 множество всех самых длинных $(1, n)$ -путей совпадает с множеством всех гамильтоновых $(1, n)$ -путей. Следовательно, справедливость неравенства (3.18) непосредственно вытекает из неравенства (3.16), что и требовалось доказать.

В графе

$$G_1(\underbrace{3,3,\dots,3}_{l-1 \text{ раз}}) \quad (n = 3 \cdot k - 1, (k \in \mathbf{N}))$$

из теоремы 3.1 множество всех самых длинных неприводимых $(1, n)$ -путей совпадает с множеством всех кратчайших $(1, n)$ -путей. Следовательно, справедливость неравенства (3.19) непосредственно вытекает из неравенства (3.7), что и требовалось доказать.

Теорема доказана.

Из теоремы 3.3, в частности, вытекает, что существуют такие законы роста $m \rightarrow \infty$, удовлетворяющие условию

$$m = \Theta(n) \quad (n \rightarrow \infty),$$

что задачи построения в явном виде как множества всех самых длинных, так и самых длинных неприводимых $(v' - v'')$ -путей $(v' \neq v'')$ в графе $G \in \mathbf{G}(n, m)$ не принадлежат классу $LCl(\mathbf{R})$ при любом представлении \mathbf{R} графа G . Тем не менее, обе эти задачи всегда принадлежат классу $WORK - LCl(\mathbf{R})$, где \mathbf{R} – представление графов, либо списками, либо матрицами смежности (либо вершин, либо ребер).

Действительно, для построения в явном виде как множества всех самых длинных, так и самых длинных неприводимых $(v' - v'')$ -путей $(v' \neq v'')$ в графе $G = (V, E)$ достаточно последовательно генерировать (одно за другим) i -размещения без повторений $(i = n - 2, n - 3, \dots)$ элементов множества $V \setminus \{v', v''\}$ (эта задача принадлежит классу $LCl(\mathbf{R})$).

Для каждой последовательности

$$\pi = [v', v_{j_1}, \dots, v_{j_i}, v'']$$

проверяется, выполнено ли свойство, соответственно, либо

« π – путь в графе $G = (V, E)$ »,

либо

« π – неприводимый путь в графе $G = (V, E)$ ».

Отметим, что проверка каждого из этих свойств – задача, принадлежащая классу $LCl(\mathbf{R})$. Если существует такое $l \in \{n-2, n-3, \dots, 1\}$, что существует последовательность

$$\pi = [v', v_{j_1}, \dots, v_{j_l}, v''],$$

обладающая проверяемым свойством, то длина искомым путей равна $l_0 + 1$, где l_0 – наибольшее из указанных значений l . Если же для всех $l \in \{n-2, n-3, \dots, 1\}$ ни одна из последовательностей

$$\pi = [v', v_{j_1}, \dots, v_{j_l}, v'']$$

не обладает проверяемым свойством, то искомое решение состоит из единственного пути $\pi = [v', v'']$, если $\{v', v''\} \in E$ и искомое решение – пустое множество, если $\{v', v''\} \notin E$.

В заключение отметим, что из полученных в настоящем пункте результатов непосредственно вытекает, что задача построения множества всех мостов для заданного графа $G = (V, E) \in \mathbf{G}(n, m)$ принадлежит классу $LCl(\mathbf{R})$, где \mathbf{R} – представление графов, либо списками, либо матрицами смежности (либо вершин, либо ребер).

Действительно, проверка того, является ли фиксированное ребро $\{v', v''\}$ мостом естественно сводится к задаче построения $(v' - v'')$ -пути длины не меньше, чем 2 и, следовательно, принадлежит классу $LCl(\mathbf{R})$, где \mathbf{R} – представление графов, либо списками, либо матрицами смежности (либо вершин, либо ребер). Для того, чтобы построить множество всех мостов для заданного графа $G \in \mathbf{G}(n, m)$ достаточно выполнить указанную проверку для каждого ребра $\{v', v''\} \in E$.

3.3. Задачи построения циклов в графе.

Ясно, что задача построения множества всех циклов в графе $G = (V, E) \in \mathbf{G}(n, m)$, проходящих через фиксированную вершину v , заведомо принадлежит классу $WORK - LCl(\mathbf{R})$.

Действительно, во-первых, каждый такой цикл представим в явном виде в рамках памяти $V(\mathbf{R}, n, m)$.

Во-вторых, проверка для последовательности

$$vv_{i_1}, \dots, v_{i_{l-1}} \quad (l = 3, \dots, |V|)$$

вершин графа $G = (V, E)$ свойства: « $vv_{i_1}, \dots, v_{i_{l-1}}$ – цикл в графе $G = (V, E)$ » осуществима в рамках памяти $V(\mathbf{R}, n, m)$.

В-третьих, последовательная генерация (одна за другой) всех последовательностей

$$vv_{i_1}, \dots, v_{i_{l-1}} \quad (l = 3, \dots, |V|)$$

попарно различных элементов множества V сводится к последовательной генерации (одного за другим) всех $(l-1)$ -размещений $(l = 3, \dots, |V|)$ без повторений

$$(v_1, \dots, v_{l-1})$$

элементов множества $V \setminus \{v\}$ (что, очевидно, осуществимо в рамках памяти $V(\mathbf{R}, n, m)$).

Рассмотрим задачи построения *кратчайших циклов* в заданном графе $G = (V, E) \in \mathbf{G}(n, m)$, проходящих через фиксированную вершину. Эти задачи являются модельными задачами теории графов и имеют многочисленные применения, как в самой теории графов, так и в ее приложениях.

Рассмотрим вначале задачу построения одного (безразлично, какого именно) кратчайшего цикла в заданном графе $G = (V, E) \in \mathbf{G}(n, m)$, проходящего через фиксированную вершину. Покажем, что эта задача принадлежит классу $LCl(\mathbf{R})$, где \mathbf{R} – представление графов, либо списками, либо матрицами смежности (либо вершин, либо ребер).

Действительно, пусть v – фиксированная вершина графа $G = (V, E)$. Найдем один (безразлично, какого именно) кратчайший цикл в графе $G = (V, E)$, проходящий через вершину v . Без ограничения общности можно считать, что $\Gamma(v) \neq \emptyset$, так как если $\Gamma(v) = \emptyset$, то, очевидно, не существует ни одного цикла, проходящего через вершину v . Для каждого $v' \in \Gamma(v)$ найдем кратчайший $(v - v')$ -путь

$$\pi_{v'} = [v, v_{i_1}, \dots, v_{i_{v'}}, v']$$

длины не меньше, чем 2 (очевидно, что эта задача принадлежит классу $LCl(\mathbf{R})$). Если указанный $(v - v')$ -путь не существует, то положим $i_{v'} = +\infty$. Положим

$$i_{\min} = \min\{i_{v'} \mid v' \in \Gamma(v)\}.$$

Если $i_{\min} = +\infty$, то в графе $G = (V, E)$ вообще не существует ни одного цикла, проходящего через вершину v . Если же $i_{\min} < +\infty$, то выберем такую вершину $v'' \in \Gamma(v)$, что $i_{v''} = i_{\min}$. Цикл $vv_{i_1}, \dots, v_{i_{v''}}v''$ является кратчайшим циклом в графе $G = (V, E)$, проходящим через вершину v .

Из приведенного выше решения, в частности, вытекает, что задача построения в явном виде множества всех кратчайших циклов в графе $G = (V, E) \in \mathbf{G}(n, m)$, проходящих через фиксированную вершину v , принадлежит классу $WORK - Lcl(\mathbf{R})$, где \mathbf{R} – представление графов, либо списками, либо матрицами смежности (либо вершин, либо ребер). Тем не менее, из следующей теоремы вытекает, что эта задача не всегда принадлежит классу $Lcl(\mathbf{R})$.

Теорема 3.4. Пусть $L_n^{кр-цикл}(G)$ ($n \in \mathbf{N}$) – максимальное число кратчайших циклов, проходящих в графе $G \in \mathbf{S}_n$ через фиксированную вершину и

$$L_n^{кр-цикл} = \max_{G \in \mathbf{S}_n} L_n^{кр-цикл}(G).$$

Если $n \geq 4$, то

$$L_n^{кр-цикл} \geq c \cdot n^3, \quad (3.20)$$

где c – некоторая положительная константа.

Доказательство. Выберем из множества

$$\mathbf{G}(n, k_1 + k_1 \cdot k_2) \quad (n \in \mathbf{N}, n \geq 4),$$

где

$$1 + k_1 + k_2 = n \quad (k_1, k_2 \in \mathbf{N}, k_1 \geq 2), \quad (3.21)$$

такой граф $G_3(k_1, k_2) = (V, E_3)$, что

$$E_3 = \{ \{1, i\} \mid i = 2, \dots, k_1 + 1 \} \cup \\ \cup \{ \{i, j\} \mid i = 2, \dots, k_1 + 1; j = k_1 + 2, \dots, 1 + k_1 + k_2 \}.$$

В графе $G_3(k_1, k_2)$ длина кратчайшего цикла, проходящего через вершину 1, равна 4, а число таких циклов равно

$$f(k_1, k_2) = \binom{k_1}{2} \cdot k_2 = \frac{1}{2} \cdot k_1 \cdot (k_1 - 1) \cdot k_2. \quad (3.22)$$

Нетрудно вычислить, что функция (3.22), при выполнении условия

$$1 + k_1 + k_2 = n \quad (k_1, k_2 > 0),$$

достигает своего наибольшего значения, если

$$k_1 = \frac{1}{3} \cdot (n + \sqrt{n^2 - 3 \cdot n + 3}),$$

$$k_2 = \frac{1}{3} \cdot (2 \cdot n - \sqrt{n^2 - 3 \cdot n + 3} - 3).$$

Это наибольшее значение функции f равно

$$\begin{aligned} & f\left(\frac{1}{3} \cdot (n + \sqrt{n^2 - 3 \cdot n + 3}), \frac{1}{3} \cdot (2 \cdot n - \sqrt{n^2 - 3 \cdot n + 3} - 3)\right) = \\ & = \frac{1}{27} \cdot (n + \sqrt{n^2 - 3 \cdot n + 3}) \cdot (n + \sqrt{n^2 - 3 \cdot n + 3} - 3) \times \\ & \quad \times (2 \cdot n - \sqrt{n^2 - 3 \cdot n + 3} - 3). \end{aligned} \quad (3.23)$$

Так как дополнительное условие $k_1, k_2 \in \mathbf{N}$ не изменяет порядок роста функции f , то из (3.23) вытекает, что

$$\begin{aligned} & f\left(\frac{1}{3} \cdot (n + \sqrt{n^2 - 3 \cdot n + 3}), \frac{1}{3} \cdot (2 \cdot n - \sqrt{n^2 - 3 \cdot n + 3} - 3)\right) = \\ & = \Theta(n^3) \quad (n \rightarrow \infty). \end{aligned} \quad (3.24)$$

Из (3.23) и (3.24) непосредственно вытекает справедливость неравенства (3.20), что и требовалось доказать.

Теорема доказана.

Из теоремы 3.4, в частности, вытекает, что существует такой закон роста $m \rightarrow \infty$, что задача построения в явном виде множества всех кратчайших циклов в графе $G = (V, E) \in \mathbf{G}(n, m)$, проходящих через фиксированную вершину, не принадлежит классу $LCI(\mathbf{R})$, где \mathbf{R} – представление графов, либо списками, либо матрицами смежности вершин.

Применительно к циклам в графе свойство «*быть не избыточным*» формализуется следующим образом.

Определение 3.2. Цикл

$$v v_{i_1}, \dots, v_{i_l}$$

в графе $G = (V, E)$ назовем *неприводимым*, если

$$(\forall j \in \{1, \dots, l-1\})(\{v, v_{i_j}\} \notin E) \& \\ \& (\forall j, h \in \{1, \dots, l\})(|j-h| \geq 2 \Rightarrow \{v_{i_j}, v_{i_h}\} \notin E). \quad (3.25)$$

Очевидно, что каждый кратчайший цикл всегда является неприводимым циклом. Это означает, что справедлива следующая оценка.

Следствие 3.2. Пусть $L_n^{\text{непр-цикл}}(G)$ ($n \in \mathbf{N}$) – максимальное число неприводимых циклов в графе $G \in \mathbf{S}_n$, проходящих через фиксированную вершину и

$$L_n^{\text{непр-цикл}} = \max_{G \in \mathbf{S}_n} L_n^{\text{непр-цикл}}(G).$$

Если $n \geq 4$, то

$$L_n^{\text{непр-цикл}} \geq c \cdot n^3, \quad (3.26)$$

где c – некоторая положительная константа.

Доказательство. В графе $G_3(k_1, k_2)$ из теоремы 3.4 множества кратчайших и неприводимых циклов, проходящих через вершину 1, совпадают. Следовательно, справедливость оценки (3.26) непосредственно вытекает из оценки (3.20).

Следствие доказано.

Из следствия 3.2, в частности, вытекает, что существует такой закон роста $m \rightarrow \infty$, что задача построения в явном виде множества всех неприводимых циклов в графе $G = (V, E) \in \mathbf{G}(n, m)$, проходящих через фиксированную вершину, не принадлежит классу $LCI(\mathbf{R})$, где \mathbf{R} – представление графов, либо списками, либо матрицами смежности вершин.

Граф $G_3(k_1, k_2)$ из теоремы 3.4 обладает тем свойством, что

$$m = \Theta(n^2) \quad (n \rightarrow \infty).$$

Докажем более сильный результат, а именно: покажем, что экспоненциальная оценка для числа неприводимых циклов в графе $G \in \mathbf{G}(n, m)$, проходящих через фиксированную вершину, достижима уже в том случае, когда

$$m = \Theta(n) \quad (n \rightarrow \infty).$$

Теорема 3.5. Пусть $L_{n,m}^{\text{непр-цикл}}(G)$ ($n \in \mathbf{N}$) – максимальное число неприводимых циклов в графе $G \in \mathbf{G}(n, m)$, проходящих через фиксированную вершину и

$$L_{n,m}^{\text{непр-цикл}} = \max_{G \in \mathbf{G}(n,m)} L_{n,m}^{\text{непр-цикл}}(G).$$

Если $n = 3 \cdot k + 1$ ($k \in \mathbf{N}$) и $n \geq 7$, то

$$L_{n,3n-6}^{\text{непр-цикл}} \geq (\sqrt[3]{3})^{n-1}, \quad (3.27)$$

Доказательство. Пусть $n = 3 \cdot k + 1$ ($k \in \mathbf{N}$) и $n \geq 7$. Выберем из множества

$$\mathbf{G}(n, k_1 + k_1 \cdot k_2 + k_2 \cdot k_3 + \dots + k_{l-2} \cdot k_{l-1} + k_{l-1}) \quad (n, l \in \mathbf{N}, l \geq 3),$$

где

$$k_1 + k_2 + \dots + k_{l-1} + 1 = n \quad (k_1, k_2, \dots, k_{l-1} \in \mathbf{N}),$$

такой граф $G_4(k_1, \dots, k_{l-1}) = (V, E_4)$, что

$$\begin{aligned} E_4 &= \{ \{1, i\} \mid i = 2, \dots, k_1 + 1 \} \cup \\ &\cup \{ \{1 + k_1 + \dots + k_{l-2} + i, 1\} \mid i = 1, \dots, k_{l-1} \} \cup \\ &\cup \left(\bigcup_{i=0}^{l-3} \{ \{1 + \sum_{j=1}^i k_j + u, 1 + \sum_{j=1}^{i+1} k_j + v\} \mid u = 1, \dots, k_{i+1}; v = 1, \dots, k_{i+2} \} \right). \end{aligned}$$

Нетрудно убедиться в том, что в графе $G_4(k_1, \dots, k_{l-1})$ число неприводимых циклов, проходящих через вершину 1 (эти циклы имеют длину, равную либо 4, либо l), равно

$$f(k_1, \dots, k_{l-1}) = \prod_{i=1}^{l-1} k_i + \frac{1}{2} \cdot k_1 \cdot (k_1 - 1) \cdot k_2.$$

Если $k_1 \geq 1$ и $k_2 > 0$, то истинно неравенство

$$\frac{1}{2} \cdot k_1 \cdot (k_1 - 1) \cdot k_2 \geq 0.$$

Следовательно, если выполнено условие $k_1, k_2, \dots, k_{l-1} \in \mathbf{N}$, то истинно неравенство

$$f(k_1, \dots, k_{l-1}) \geq \prod_{i=1}^{l-1} k_i. \quad (3.28)$$

Так как $n = 3 \cdot k + 1$ ($k \in \mathbf{N}$), $n \geq 7$ и

$$k_1 + k_2 + \dots + k_{l-1} + 1 = n,$$

то, положив в (3.28)

$$k_1 = k_2 = \dots = k_{l-1} = 3,$$

получим

$$f(\underbrace{3, 3, \dots, 3}_{l-1 \text{ раз}}) \geq (\sqrt[3]{3})^{n-1}. \quad (3.29)$$

Из (3.29) непосредственно вытекает справедливость неравенства (3.27), что и требовалось доказать.

Теорема доказана.

Замечание 3.8. Граф $G_4(k_1, \dots, k_{l-1})$ ($l \in \mathbf{N}, l \geq 3$) из теоремы 3.5 обладает тем свойством, что для каждого числа

$$i \in \{4, 6, \dots, 2 \cdot (l-1)\} \cup \{l\}$$

существует цикл длины i , проходящий через вершину 1. Однако среди этих циклов неприводимыми являются только циклы длин 4 и l .

Из теоремы 3.5, в частности, вытекает, что существует такой закон роста $m \rightarrow \infty$, удовлетворяющий условию

$$m = \Theta(n) \quad (n \rightarrow \infty),$$

что задача построения в явном виде множества всех неприводимых циклов, проходящих через фиксированную вершину $v \in V$ графа $G = (V, E) \in \mathbf{G}(n, m)$, не принадлежит классу $Lcl(\mathbf{R})$ при любом представлении \mathbf{R} графа G . Тем не менее, эта задача принадлежит классу $WORK-Lcl(\mathbf{R})$, где \mathbf{R} – представление графов, либо списками, либо матрицами смежности (либо вершин, либо ребер).

Действительно, достаточно последовательно (одну за другой) генерировать все последовательности

$$vv_{i_1}, \dots, v_{i_{l-1}} \quad (l = 3, \dots, |V|)$$

попарно различных элементов множества V (эта задача принадлежит классу $LCl(\mathbf{R})$) и проверять для каждой из них, выполнено ли свойство:

$$\ll vv_{i_1}, \dots, v_{i_{l-1}} - \text{неприводимый цикл в графе } G = (V, E) \gg.$$

Отметим, что проверка этого свойства является задачей, принадлежащей классу $LCl(\mathbf{R})$.

Рассмотрим теперь задачу построения *гамильтоновых* циклов в заданном графе $G = (V, E) \in \mathbf{G}(n, m)$. Эти задачи являются модельными задачами теории графов и имеют многочисленные применения, как в самой теории графов, так и в ее приложениях.

Задача построения в явном виде одного (безразлично, какого именно) гамильтонового цикла в графе $G = (V, E) \in \mathbf{G}(n, m)$ принадлежит классу $LCl(\mathbf{R})$, где \mathbf{R} – представление графов, либо списками, либо матрицами смежности (либо вершин, либо ребер).

Действительно, достаточно зафиксировать вершину $v \in V$ и последовательно генерировать (одно за другим) $(n - 1)$ -размещения без повторений элементов множества $V \setminus \{v\}$, каждый раз сохраняя только текущее размещение (очевидно, что эта задача принадлежит классу $LCl(\mathbf{R})$). Для каждой последовательности

$$vv_{i_1}, \dots, v_{i_{n-1}}$$

проверяется, выполнено ли свойство:

$$\ll vv_{i_1}, \dots, v_{i_{n-1}} - \text{цикл в графе } G = (V, E) \gg.$$

При положительном ответе происходит остановка вычислений, так как искомый гамильтонов цикл найден. Если осуществлена проверка всех указанных выше последовательностей и положительный ответ не найден, то искомый путь не существует.

Очевидно, что задача построения множества всех гамильтоновых циклов в графе $G = (V, E) \in \mathbf{G}(n, m)$ не принадлежит классу $LCl(\mathbf{R})$ при любом представлении \mathbf{R} графа G .

Действительно, известно, что число гамильтоновых циклов в полном графе K_n равно $(n-1)!$. Как было отмечено ранее, K_n – экстремальный граф (так как $K_n \in \mathbf{G}(n, \binom{n}{2})$).

Покажем, что экспоненциальная оценка для числа гамильтоновых циклов в графе $G \in \mathbf{G}(n, m)$, достижима уже в случае, когда

$$m = \Theta(n) \quad (n \rightarrow \infty).$$

Теорема 3.6. Пусть $L_{n,m}^{\text{гам-цикл}}(G)$ ($n, m \in \mathbf{N}, 0 \leq m \leq \binom{n}{2}$) – максимальное число гамильтоновых циклов в графе $\mathbf{G}(n, m)$ и

$$L_{n,m}^{\text{гам-цикл}} = \max_{G \in \mathbf{G}(n,m)} L_{n,m}^{\text{гам}}(G).$$

Если $n = 3 \cdot k + 2$ ($k \in \mathbf{N}$), то

$$L_{n,4 \cdot n - 10}^{\text{гам-цикл}} \geq 0.5 \cdot (\sqrt[3]{12})^{n-2}. \quad (3.30)$$

Доказательство. Пусть $n \equiv 2 \pmod{3}$ и $n \geq 5$. Рассмотрим граф

$$G_5(\underbrace{3, 3, \dots, 3}_{\frac{n-2}{3} \text{ раз}}) = (V, E_5) \in \mathbf{G}(n, 4 \cdot n - 10),$$

полученный в результате следующей небольшой модификации графа

$$G_2(\underbrace{3, 3, \dots, 3}_{\frac{n-2}{3} \text{ раз}}) = (V, E_2)$$

из теоремы 3.2:

$$E_5 = E_2 \cup \{1, n\}.$$

Каждому гамильтоновому $(1, n)$ -пути

$$\pi = [1, i_1, \dots, i_{n-2}, n]$$

в графе $G_2(\underbrace{3, 3, \dots, 3}_{\frac{n-2}{3} \text{ раз}})$ можно поставить в соответствие гамильтонов цикл

$$1i_1 \dots i_{n-2}n$$

в графе $G_5(\underbrace{3,3,\dots,3}_{\frac{n-2}{3} \text{ раз}})$. Очевидно, что указанное соответствие является однозначным, т.е. различным гамильтоновым $(1, n)$ -путям в графе $G_2(\underbrace{3,3,\dots,3}_{\frac{n-2}{3} \text{ раз}})$ соответствуют различные гамильтоновы циклы в графе $G_5(\underbrace{3,3,\dots,3}_{\frac{n-2}{3} \text{ раз}})$. Отсюда непосредственно вытекает, что число гамильтоновых циклов в графе $G_5(\underbrace{3,3,\dots,3}_{\frac{n-2}{3} \text{ раз}})$ не меньше, чем число гамильтоновых $(1, n)$ -путей в графе $G_2(\underbrace{3,3,\dots,3}_{\frac{n-2}{3} \text{ раз}})$. Следовательно, справедливость неравенства (3.30) вытекает из неравенства (3.16), что и требовалось доказать.

Теорема доказана.

Из теоремы 3.6, в частности, вытекает, что существует такой закон роста $m \rightarrow \infty$, удовлетворяющий условию

$$m = \Theta(n) \quad (n \rightarrow \infty),$$

что задача построения в явном виде множества всех гамильтоновых циклов в графе $G \in \mathbf{G}(n, m)$ не принадлежит классу $LCl(\mathbf{R})$ при любом представлении \mathbf{R} графа G . Тем не менее, эта задача всегда принадлежит классу $WORK - LCl(\mathbf{R})$, где \mathbf{R} – представление графов, либо списками, либо матрицами смежности (либо вершин, либо ребер).

Действительно, достаточно зафиксировать вершину v графа $G = (V, E) \in \mathbf{G}(n, m)$ и генерировать последовательно (одну за другой) последовательности

$$vv_{i_1}, \dots, v_{i_{n-1}},$$

попарно различных элементов множества V , проверяя для каждой из них, выполнено ли свойство:

$$\ll vv_{i_1}, \dots, v_{i_{n-1}} \text{ – гамильтонов цикл в графе } G = (V, E) \gg.$$

В заключение рассмотрим задачи построения в явном виде множеств, как всех самых длинных, так и всех самых длинных неприводимых циклов, проходящих через фиксированную вершину $v \in V$ графа $G = (V, E) \in \mathbf{G}(n, m)$. Справедливы следующие оценки.

Теорема 3.7. Пусть $L_{n,m}^{\max\text{-цикл}}(G)$ и $L_{n,m}^{\max\text{-непр-цикл}}(G)$ ($n, m \in \mathbf{N}, 0 \leq m \leq \binom{n}{2}$)

– максимальное число, соответственно, самых длинных и самых длинных неприводимых циклов, проходящих через фиксированную вершину графа $G \in \mathbf{G}(n, m)$,

$$L_{n,m}^{\max\text{-цикл}} = \max_{G \in \mathbf{G}(n,m)} L_{n,m}^{\max\text{-цикл}}(G)$$

и

$$L_{n,m}^{\max\text{-непр-цикл}} = \max_{G \in \mathbf{G}(n,m)} L_{n,m}^{\max\text{-непр-цикл}}(G).$$

Тогда:

1. Если $n = 3 \cdot k + 2$ ($k \in \mathbf{N}$), то

$$L_{n,4 \cdot n - 10}^{\max\text{-цикл}} \geq 0.5 \cdot (\sqrt[3]{12})^{n-2}. \quad (3.31)$$

2. Если $n \equiv 1 \pmod{3}$ и $n \geq 13$, то

$$L_{n,3 \cdot n - 6}^{\max\text{-непр-цикл}} \geq (\sqrt[3]{3})^{n-1}. \quad (3.32)$$

Доказательство. 1. В графе

$$G_5(\underbrace{3,3,\dots,3}_{\frac{n-2}{3} \text{ раз}}) = (V, E_5) \in \mathbf{G}(n, 4 \cdot n - 10) \quad (n = 3 \cdot k + 2, (k \in \mathbf{N}))$$

из теоремы 3.6, множество всех самых длинных циклов, проходящих через любую фиксированную вершину, совпадает с множеством всех гамильтоновых циклов. Следовательно, справедливость неравенства (3.31) непосредственно вытекает из неравенства (3.30), что и требовалось доказать.

Рассмотрим граф

$$G_4(\underbrace{3,3,\dots,3}_{\frac{n-1}{3} \text{ раз}}) = (V, E_4) \in \mathbf{G}(n, 3 \cdot n - 6) \quad (n = 3 \cdot k + 1, (k \in \mathbf{N}))$$

из теоремы 3.5. Если $n \geq 13$, то множество всех самых длинных неприводимых циклов, проходящих через вершину 1, совпадает с множеством всех неприводимых циклов длины l , проходящих через вершину 1 (см. замечание 3.8). Следовательно, справедливость неравенства (3.32) непосредственно вытекает из неравенства (3.27) (см. доказательство теоремы 3.5), что и требовалось доказать.

Теорема доказана.

Из теоремы 3.7, в частности, вытекает, что существуют такие законы роста $m \rightarrow \infty$, удовлетворяющие условию

$$m = \Theta(n) \quad (n \rightarrow \infty),$$

что задачи построения в явном виде как множества всех самых длинных, так и самых длинных неприводимых циклов, проходящих через фиксированную вершину графа $G \in \mathbf{G}(n, m)$, не принадлежат классу $LCI(\mathbf{R})$ при любом представлении \mathbf{R} графа G . Тем не менее, обе эти задачи всегда принадлежат классу $WORK - LCI(\mathbf{R})$, где \mathbf{R} – представление графов, либо списками, либо матрицами смежности (либо вершин, либо ребер).

Действительно, для построения в явном виде как множества всех самых длинных, так и самых длинных неприводимых циклов, проходящих через фиксированную вершину $v \in V$ графа $G \in \mathbf{G}(n, m)$, достаточно последовательно генерировать (одно за другим) i -размещения без повторений ($i = n-1, n-2, \dots$) элементов множества $V \setminus \{v\}$ (эта задача принадлежит классу $LCI(\mathbf{R})$). Для каждой последовательности

$$vv_{j_1}, \dots, v_{j_i}$$

проверяется, выполнено ли свойство, соответственно, либо

$$\langle\langle vv_{j_1}, \dots, v_{j_i} - \text{цикл в графе } G = (V, E) \rangle\rangle,$$

либо

$$\langle\langle vv_{j_1}, \dots, v_{j_i} - \text{неприводимый цикл в графе } G = (V, E) \rangle\rangle.$$

Отметим, что проверка каждого из этих свойств – задача, принадлежащая классу $LCI(\mathbf{R})$. Если существует такое $l \in \{n-1, n-2, \dots, 3\}$, что существует последовательность

$$vv_{j_1}, \dots, v_{j_l},$$

обладающая проверяемым свойством, то длина искомых циклов равна $l_0 + 1$, где l_0 – наибольшее из указанных значений l . Если же для всех $l \in \{n-1, n-2, \dots, 3\}$ ни одна из последовательностей

$$\pi = [v', v_{j_1}, \dots, v_{j_l}, v'']$$

не обладает проверяемым свойством, то искомое решение – пустое множество.

3.4. Построение остовных деревьев.

Всюду в настоящем пункте считаем, что $G = (V, E) \in \mathbf{S}_n$ – связный граф (ранее было показано, что проверка свойства «быть связным» для графа $G = (V, E) \in \mathbf{S}_n$ – задача, принадлежащая классу $LCl(\mathbf{R})$, где \mathbf{R} – представление графов, либо списками, либо матрицами смежности (либо вершин, либо ребер)). Отметим, что из свойства «быть связным» непосредственно вытекает, что если $G = (V, E) \in \mathbf{G}(n, m)$ ($n \in \mathbf{N}$) – связный граф, то $m \geq n - 1$. Обозначим через $\mathbf{G}_{con}(n, m)$ множество всех связных графов $G = (V, E) \in \mathbf{G}(n, m)$ ($m \geq n - 1$).

Покажем, что задача построения множества всех остовных деревьев для графа $G = (V, E) \in \mathbf{G}_{con}(n, m)$ ($m \geq n - 1$) заведомо принадлежит классу $WORK - LCl(\mathbf{R})$, где \mathbf{R} – представление графов, либо списками, либо матрицами смежности (либо вершин, либо ребер).

Действительно, каждое остовное дерево графа $G = (V, E) \in \mathbf{G}_{con}(n, m)$ представимо в явном виде в рамках памяти $V(\mathbf{R}, n, m)$, так как оно является графом, принадлежащим множеству $\mathbf{G}_{con}(n, n - 1)$. Поэтому, достаточно пронумеровать ребра графа G числами $1, 2, \dots, m$ (т.е., фактически, построить таблицу 1.1 для графа G , что является задачей, принадлежащей классу $LCl(\mathbf{R})$) и затем последовательно (одно за другим) генерировать $(n - 1)$ -размещения без повторений

$$i_1 i_2 \dots i_{n-1}$$

элементов множества $\{1, 2, \dots, m\}$ (что осуществимо в рамках памяти $V(\mathbf{R}, n, m)$). Каждая порожденная последовательность $i_1 i_2 \dots i_{n-1}$ определяет подграф $G_{i_1, i_2, \dots, i_{n-1}}$ графа G с множеством ребер $\{i_1, i_2, \dots, i_{n-1}\}$ (очевидно, что этот подграф представим в рамках памяти $V(\mathbf{R}, n, m)$). Для графа $G_{i_1, i_2, \dots, i_{n-1}}$ осуществляется проверка свойства:

«быть порядка n и быть связным».

Нетрудно убедиться в том, что проверка этого свойства осуществима в рамках памяти $V(\mathbf{R}, n, m)$. В случае положительного ответа граф $G_{i_1, i_2, \dots, i_{n-1}}$ – очередное остовное дерево.

Отметим, что из приведенного выше решения непосредственно вытекает, что задача построения одного (безразлично, какого именно) остовного дерева для графа $G = (V, E) \in \mathbf{G}_{con}(n, m)$ принадлежит классу $LCl(\mathbf{R})$, где \mathbf{R} – представление графов, либо списками, либо матрицами смежности (либо вершин, либо ребер).

Покажем, что существует такой закон роста $m \rightarrow \infty$, удовлетворяющие условию

$$m = \Theta(n) \quad (n \rightarrow \infty),$$

что задачи построения в явном виде множества всех остовных деревьев для графа $G = (V, E) \in \mathbf{G}_{con}(n, m)$ не принадлежит классу $LCI(\mathbf{R})$ при любом представлении \mathbf{R} графа G .

Следствие 3.3. Пусть $L_n^{остовн-дер}(G)$ ($n \in \mathbf{N}$) – максимальное число остовных деревьев в графе $G \in \mathbf{G}_{con}(n, m)$ и

$$L_n^{остовн-дер} = \max_{G \in \mathbf{G}_{con}(n, m)} L_n^{остовн-дер}(G).$$

Если $n = 3 \cdot k + 2$ ($k \in \mathbf{N}$), то

$$L_{n, 4 \cdot n - 11}^{остовн-дер} \geq 0.5 \cdot (\sqrt[3]{12})^{n-2}. \quad (3.33)$$

Доказательство. Любой гамильтонов путь в графе является остовным деревом для этого графа. Предположим, что существует гамильтонов $(v' - v'')$ -путь ($v' \neq v''$) между двумя фиксированными вершинами v' и v'' графа $G \in \mathbf{G}_{con}(n, m)$. Тогда число остовных деревьев для графа G не меньше, чем число гамильтоновых $(v' - v'')$ -путей в графе G . Теперь достаточно применить теорему 3.2, чтобы убедиться в том, что существует граф $G \in \mathbf{G}_{con}(n, 4 \cdot n - 11)$, для которого число остовных деревьев удовлетворяет неравенству (3.33), что и требовалось доказать.

Следствие доказано.

Из следствия 3.3 непосредственно вытекает, что существует такой закон роста $m \rightarrow \infty$, удовлетворяющие условию

$$m = \Theta(n) \quad (n \rightarrow \infty),$$

что задачи построения в явном виде множества всех остовных деревьев для графа $G = (V, E) \in \mathbf{G}_{con}(n, m)$ не принадлежит классу $LCI(\mathbf{R})$ при любом представлении \mathbf{R} графа G .

Гамильтонов $(v' - v'')$ -путь $\pi_{v'-v''}$ в графе $G = (V, E)$ (если он существует) является остовным деревом, удовлетворяющим следующему условию: существует хотя бы одна такая вершина v графа G , что $(v' - v)$ -путь в дереве $\pi_{v'-v''}$ является наиболее длинным $(v' - v)$ -путем в графе G .

Замечание 3.9. Указанному выше свойству в остовном дереве $\pi_{v'-v''}$, например, всегда удовлетворяет вершина v'' .

Важный класс (непустой для любого связного графа $G = (V, E)$) образуют следующие остовные деревья.

Определение 3.3. Остовное дерево T графа $G = (V, E)$ называется *деревом кратчайших путей* из вершины $v' \in V$, если $(v' - v'')$ -путь в дереве T является кратчайшим $(v' - v'')$ -путем в графе G для любой вершины $v'' \in V \setminus \{v'\}$.

Задача построения для графа $G = (V, E) \in \mathbf{G}_{con}(n, m)$ одного (безразлично, какого именно) дерева кратчайших путей из фиксированной вершины $v' \in V$ принадлежит классу $LCl(\mathbf{R})$, где \mathbf{R} – представление графов, либо списками, либо матрицами смежности (либо вершин, либо ребер).

Действительно, во-первых, достаточно построить (по аналогии с тем, как это было сделано в алгоритме 3.1) такую конечную последовательность множеств

$$U_0, U_1, \dots, U_k, U_{k+1},$$

что

$$U_0 = \{v'\}, \quad U_{k+1} = \emptyset$$

и

$$U_{i+1} = \Gamma(U_i) \setminus \left(\bigcup_{j=0}^i U_j \right) \quad (i = 0, 1, \dots, k).$$

Во-вторых, для каждой вершины v'' , принадлежащей множеству U_{i+1} ($i = k-1, k-2, \dots, 1, 0$) достаточно выбрать одно (безразлично, какое именно) ребро $\{v'', v\}$, где $v \in U_i$.

Из возможности применения поиска с возвращением (*Backtracking*) для организации выбора ребер

$$\{v, v''\} \quad (v'' \in U_{i+1}, v \in U_i)$$

всеми возможными способами непосредственно вытекает, что для графа $G = (V, E) \in \mathbf{G}_{con}(n, m)$ задача построения множества всех деревьев кратчайших путей из фиксированной вершины $v' \in V$ заведомо принадлежит классу $WORK - LCl(\mathbf{R})$, где \mathbf{R} – представление графов, либо списками, либо матрицами смежности (либо вершин, либо ребер).

Покажем, что существует такой закон роста $m \rightarrow \infty$, что эта задача не принадлежит классу $LC/(\mathbf{R})$ при любом представлении \mathbf{R} графа $G = (V, E) \in \mathbf{G}_{con}(n, m)$.

Теорема 3.8. Пусть $L_{n,m}^{\text{дер-кр-пут}}(G)$ ($n, m \in \mathbf{N}, n-1 \leq m \leq \binom{n}{2}$) – максимальное число деревьев кратчайших путей из фиксированной вершины графа $G \in \mathbf{G}_{con}(n, m)$ и

$$L_{n,m}^{\text{дер-кр-пут}} = \max_{G \in \mathbf{G}_{con}(n,m)} L_{n,m}^{\text{дер-кр-пут}}(G).$$

Тогда:

1. Если $n = 3 \cdot k + 1$ ($k \in \mathbf{N}$), то

$$L_{n,3 \cdot n-9}^{\text{дер-кр-пут}} \geq 3^{n-4}. \quad (3.34)$$

2. Если $n = 3 \cdot k + 1$ ($k \in \mathbf{N}$) и $n \geq 7$, то

$$L_{n,0.25 \cdot (n^2-1)}^{\text{дер-кр-пут}} \geq (\sqrt{0.5 \cdot (n-1)})^{n-1}. \quad (3.35)$$

Доказательство. Пусть $n = 3 \cdot k + 1$ ($k \in \mathbf{N}$). Выберем из множества

$$\mathbf{G}_{con}(n, k_1 + k_1 \cdot k_2 + k_2 \cdot k_3 + \dots + k_{l-2} \cdot k_{l-1}) \quad (n, l \in \mathbf{N}, l \geq 2),$$

где

$$k_1 + k_2 + \dots + k_{l-1} + 1 = n \quad (k_1, k_2, \dots, k_{l-1} \in \mathbf{N}), \quad (3.36)$$

такой граф $G_6(k_1, \dots, k_{l-1}) = (V, E_6)$, что

$$E_6 = \{ \{1, i\} \mid i = 2, \dots, k_1 + 1 \} \cup \\ \cup \left(\bigcup_{i=0}^{l-3} \{ \{1 + \sum_{j=1}^i k_j + u, 1 + \sum_{j=1}^{i+1} k_j + v\} \mid u = 1, \dots, k_{i+1}; v = 1, \dots, k_{i+2} \} \right).$$

Покажем, что в графе $G_6(k_1, \dots, k_{l-1})$ число деревьев кратчайших путей из вершины 1 равно

$$f(k_1, \dots, k_{l-1}) = k_1^{k_2} \cdot k_2^{k_3} \cdot k_{l-2}^{k_{l-1}}. \quad (3.37)$$

Действительно, в графе $G_6(k_1, \dots, k_{l-1})$ любое дерево T кратчайших путей из вершины 1 содержит все ребра, принадлежащие множеству

$$\{ \{1, i\} \mid i = 2, \dots, k_1 + 1 \},$$

а также, для каждого значения $i = 0, 1, \dots, l-3$ в точности k_{i+2} ребер, соединяющих вершины, принадлежащие множеству

$$\{1 + \sum_{j=1}^{i+1} k_j + v \mid v = 1, \dots, k_{i+2}\}$$

с вершинами, принадлежащими множеству

$$\{1 + \sum_{j=1}^i k_j + u \mid u = 1, \dots, k_{i+1}\}.$$

Выбор указанных выше k_{i+2} ($i = 0, 1, \dots, l-3$) ребер осуществим $k_{i+1}^{k_{i+2}}$ способами (что совпадает с числом отображений k_{i+2} -элементного множества в k_{i+1} -элементное множество). Для различных значений $i = 0, 1, \dots, l-3$ выбор ребер осуществляется независимо, откуда и вытекает справедливость формулы (3.37).

Рассмотрим граф

$$G_6(\underbrace{3, 3, \dots, 3}_{\frac{n-1}{3} \text{ раз}}) = (V, E_6) \in \mathbf{G}_{con}(n, 3 \cdot n - 9).$$

Положим в (3.36)

$$k_1 = k_2 = \dots = k_{l-1} = 3. \quad (3.38)$$

Получим

$$3 \cdot (l-1) + 1 = n \Rightarrow l-2 = \frac{n-4}{3}. \quad (3.39)$$

Подставим (3.38) и (3.39) в (3.37). Получим

$$f(\underbrace{3, 3, \dots, 3}_{\frac{n-1}{3} \text{ раз}}) = \underbrace{3^3 \cdot \dots \cdot 3^3}_{\frac{n-4}{3} \text{ раз}} = 27^{\frac{n-4}{3}} = 3^{n-4}. \quad (3.40)$$

Из равенства (3.40) непосредственно вытекает, что неравенство (3.34) – истинное, что и требовалось доказать.

Рассмотрим граф

$$G_6(k, k) = (V, E_6) \in \mathbf{G}_{con}(n, k + k^2).$$

Положим в (3.36) $l = 3$ и

$$k_1 = k_2 = k. \quad (3.41)$$

Получим

$$2 \cdot k + 1 = n \Rightarrow k = 0.5 \cdot (n - 1). \quad (3.42)$$

Подставим (3.41) и (3.42) в (3.37). Получим

$$f(0.5 \cdot (n - 1), 0.5 \cdot (n - 1)) = (\sqrt{0.5 \cdot (n - 1)})^{n-1}. \quad (3.43)$$

Из равенства (3.43) непосредственно вытекает, что неравенство (3.35) – истинное, что и требовалось доказать.

Теорема доказана.

Из теоремы 3.8, в частности, вытекает, что существует такой закон роста $m \rightarrow \infty$, удовлетворяющий условию

$$m = \Theta(n) \quad (n \rightarrow \infty),$$

что задача построения в явном виде множества всех деревьев кратчайших путей из фиксированной вершины $v' \in V$ графа $G = (V, E) \in \mathbf{G}_{con}(n, m)$, не принадлежит классу $LCl(\mathbf{R})$ при любом представлении \mathbf{R} графа G .

Рассмотрим теперь задачи построения экономных остовных деревьев. Эти деревья образуют один из основных классов (непустой для любого связного графа $G = (V, E)$) остовных деревьев и имеют многочисленные применения в процессе разработки и анализа алгоритмов на графах.

Пусть на множестве вершин графа $G = (V, E) \in \mathbf{G}_{con}(n, m)$ зафиксирована функция стоимости ребер

$$COST_G : E \rightarrow \mathbf{Z}_+. \quad (3.44)$$

Обозначим через $\mathbf{ST}(G)$ ($G \in \mathbf{G}_{con}(n, m)$) множество всех остовных деревьев графа G . Распространим функцию $COST_G$ на множество $\mathbf{ST}(G)$ следующим образом: если $T = (V, E_T) \in \mathbf{ST}(G)$, то

$$COST_G(T) = \sum_{e \in E_T} COST(e). \quad (3.45)$$

Определение 3.4. Остовное дерево $T \in \mathbf{ST}(G)$ графа $G \in \mathbf{G}_{con}(n, m)$ называется *экономным (economical spanning tree)* (или, иными словами, *остовным деревом наименьшей стоимости*), если

$$COST_G(T) = \min_{T' \in \mathbf{ST}(G)} COST_G(T'). \quad (3.46)$$

Обозначим через $\mathbf{ST}_{econ}(G)$ ($G \in \mathbf{G}_{con}(n, m)$) множество всех экономных остовных деревьев графа G .

Задача построения одного (безразлично, какого именно) экономного дерева $T \in \mathbf{ST}_{econ}(G)$ принадлежит классу $LCl(\mathbf{R})$, где \mathbf{R} – представление графов, либо списками, либо матрицами смежности (либо вершин, либо ребер).

Действительно (см., напр. [14]), чтобы построить экономное дерево $T = (V, E_T) \in \mathbf{ST}_{econ}(G)$ графа $G = (V, E) \in \mathbf{G}_{con}(n, m)$, достаточно последовательно, одно за другим, так выбирать из множества оставшихся ребер графа G самые дешевые (т.е. для них значение функции $COST_G$ – наименьшее), чтобы множество выбранных ребер не содержало циклов. Соответствующий алгоритм имеет следующий вид.

Алгоритм 3.6.

Шаг 1. $E_T := \emptyset$, $T := (V, E_T)$, $U := E$, $i := 1$.

Шаг 2.

$$Del(U) := \{e \in U \mid \text{граф } (V, E_T \cup \{e\}) \text{ содержит цикл}\}.$$

Шаг 3. $U := U \setminus Del(U)$.

Шаг 4.

$$CHEAPEST(U) := \{e \in U \mid COST_G(e) = \min_{e' \in U} COST_G(e')\}.$$

Шаг 5. Выбрать ребро $e \in CHEAPEST(U)$, $u_i := e$, $E_T := E_T \cup \{u_i\}$, $U := U \setminus \{u_i\}$.

Шаг 6. $i := i + 1$.

Шаг 7. Если $i \leq |V| - 1$, то переход к шагу 2, иначе *конец*.

Корректность алгоритма 3.6, фактически, показана в [14, стр. 13].

Нетрудно убедиться в том, что емкостная сложность алгоритма 3.6 равна $V(\mathbf{R}, n, m)$. Отсюда непосредственно вытекает, что задача построения одного (безразлично, какого именно) экономного дерева $T \in \mathbf{ST}_{econ}(G)$ принадлежит классу $LCl(\mathbf{R})$, где \mathbf{R} – представление графов, либо списками, либо матрицами смежности (либо вершин, либо ребер), что и требовалось показать.

Из возможности применения поиска с возвратом (*Backtracking*) для организации выбора ребер $e \in CHEAPEST(U)$ всеми возможными способами вытекает, что для графа $G = (V, E) \in \mathbf{G}_{con}(n, m)$ задача построения множества $\mathbf{ST}_{econ}(G)$ всех экономных остовных деревьев заведомо принад-

лежит классу $WORK - LCl(\mathbf{R})$, где \mathbf{R} – представление графов, либо списками, либо матрицами смежности (либо вершин, либо ребер). Тем не менее, для любого представления \mathbf{R} графа G эта задача не всегда принадлежит классу $LCl(\mathbf{R})$.

Действительно, достаточно рассмотреть специальный случай, когда функция стоимости ребер имеет следующий вид

$$COST_G(e) = 1 \quad (G = (V, E) \in \mathbf{G}_{con}(n, m))$$

для всех $e \in E$. Теперь достаточно применить следствие 3.3. Более того, из теоремы 3.8 непосредственно вытекает, что для рассматриваемого случая справедливо следующее следствие.

Следствие 3.4. Пусть $L_{n,m}^{экон-ост-дер}(G)$ ($n, m \in \mathbf{N}, n-1 \leq m \leq \binom{n}{2}$) – максимальное число экономных остовных деревьев в графе $G \in \mathbf{G}_{con}(n, m)$, где максимум берется по всем функция стоимости ребер. Положим

$$L_{n,m}^{экон-ост-дер} = \max_{G \in \mathbf{G}_{con}(n,m)} L_{n,m}^{экон-ост-дер}(G).$$

Тогда:

1. Если $n = 3 \cdot k + 1$ ($k \in \mathbf{N}$), то

$$L_{n,3 \cdot n-9}^{экон-ост-дер} \geq 3^{n-4}. \quad (3.47)$$

2. Если $n = 3 \cdot k + 1$ ($k \in \mathbf{N}$) и $n \geq 7$, то

$$L_{n,0.25 \cdot (n^2-1)}^{экон-ост-дер} \geq (\sqrt{0.5 \cdot (n-1)})^{n-1}. \quad (3.48)$$

В заключение отметим, что из полученных в настоящем пункте результатов непосредственно вытекает принадлежность классам $LCl(\mathbf{R})$ и $WORK - LCl(\mathbf{R})$ задач, связанных с построением для заданного графа $G \in \mathbf{G}_{con}(n, m)$, соответственно, одного (безразлично, какого именно) и всех кодеревьев, т.е. дополнений к остовным деревьям. Отсюда вытекает, что классам $LCl(\mathbf{R})$ и $WORK - LCl(\mathbf{R})$ принадлежат задачи построения для заданного графа $G \in \mathbf{G}_{con}(n, m)$, соответственно, одного (безразлично, какого именно) и всех коциклов. В частности, классу $LCl(\mathbf{R})$ принадлежит задача построения базиса коциклов, а классу $WORK - LCl(\mathbf{R})$ – задача построения всех базисов коциклов для заданного графа $G \in \mathbf{G}_{con}(n, m)$.

3.5. Pebbling.

Составной частью решения достаточно широкого класса задач, сформулированных в терминах графов, является построение обхода вершин заданного графа в соответствии с теми или иными правилами. Традиционно в теории графов и ее приложениях исследуются обходы вершин графов, связанные с построением либо гамильтонового пути или цикла, либо эйлерового контура, следа или прогулки (см., напр., [5]). Другой тип обходов для специальных (возможно, бесконечных) графов с выделенными вершинами – прямоугольных лабиринтов (см., напр., [6]) – является предметом изучения теории алгоритмов, представленной в терминах МТ и связан, в основном, с исследованием алгоритмической разрешимости задачи построения требуемого обхода. В настоящем пункте рассматривается совершенно иной тип обхода вершин заданного конечного связанного ациклического орграфа. Этот тип обхода базируется на аксиоматике *теории сетевого планирования и управления (PERT)* и характеризуется тем, что «*наступление любого очередного этапа возможно только после того, как наступили все непосредственно предшествующие ему этапы*». Именно согласованность рассматриваемых в настоящем пункте обходов с аксиоматикой *PERT* является гарантией их широкого применения в процессе решения прикладных задач. Введем необходимые понятия и определения.

Обозначим через $\vec{\mathbf{G}}_{con-acycl}^{or}(n, m)$ множество всех связных ациклических орграфов $G \in \vec{\mathbf{G}}^{or}(n, m)$. Пусть в орграфе $G = (V, E) \in \vec{\mathbf{G}}_{con-acycl}^{or}(n, m)$ зафиксированы две такие вершины v' и v'' ($v', v'' \in V; v' \neq v''$), что v' – источник (т.е. $\Gamma^{-1}(v') = \emptyset$). Процесс $(v' \rightarrow v'')$ -pebbling ($\Gamma^{-1}(v') = \emptyset$) орграфа $G \in \vec{\mathbf{G}}_{con-acycl}^{or}(n, m)$ определяется как любой процесс обхода вершин орграфа G в моменты времени $t \in \mathbf{Z}_+$ в соответствии со следующими правилами:

1. В начальный момент времени $t = 0$ камешком покрывается вершина v' .
2. В момент времени $t \in \mathbf{N}$ вершина $v \in V$ может быть покрыта камешком тогда и только тогда, когда в момент времени $t - 1$ покрыты камешками все вершины, принадлежащие множеству $\Gamma^{-1}(v)$.
3. Камешек может быть убран в любой момент времени с любой вершины $v \in V$, покрытой в этот момент камешком.
4. Процесс *успешно завершается (terminates)* в первый момент времени, когда камешком покрывается вершина v'' .
5. Процесс – *неуспешный (fails)*, если не существует момент времени $t \in \mathbf{N}$, когда вершина v'' покрыта камешком.

Задача, состоящая в том, чтобы выяснить существует ли (успешный) процесс $(v' \rightarrow v'')$ -pebbling ($\Gamma^{-1}(v') = \emptyset$) для орграфа $G \in \vec{\mathbf{G}}_{con-acycl}^{or}(n, m)$ за-

ведомо принадлежит классу $LcI(\mathbf{R})$, где \mathbf{R} – представление графов, либо списками, либо матрицами смежности (либо вершин, либо ребер). Действительно, эта задача имеет положительное решение тогда и только тогда, когда орграф $G \in \overrightarrow{\mathbf{G}}_{con-acycl}^{or}(n, m)$ удовлетворяет следующим двум условиям: 1) существует путь, идущий из вершины v' в вершину v'' ; 2) не существует такой вершины v , что существует путь, идущий из вершины v в вершину v'' и не существует путь, идущий из вершины v' в вершину v .

Ясно, что в случае положительного ответа, задача, состоящая в построении одного (безразлично, какого именно) процесса $(v' \rightarrow v'')$ -pebbling ($\Gamma^{-1}(v') = \emptyset$) для орграфа $G \in \overrightarrow{\mathbf{G}}_{con-acycl}^{or}(n, m)$, также заведомо принадлежит классу $LcI(\mathbf{R})$. Тривиальное решение этой задачи легко получить с помощью $|V|$ камешков.

Как правило, использование камешка для разметки вершины в теоретико-графовой модели связано с материальными «затратами» в реальном моделируемом процессе. Поэтому естественно рассматривать только процесс $(v' \rightarrow v'')$ -pebbling ($\Gamma^{-1}(v') = \emptyset$) для орграфа $G \in \overrightarrow{\mathbf{G}}_{con-acycl}^{or}(n, m)$, осуществляемый с помощью минимально возможного числа камешков. Это наименьшее число камешков, в конечном итоге, характеризует сложность процесса $(v' \rightarrow v'')$ -pebbling ($\Gamma^{-1}(v') = \emptyset$) для орграфа $G \in \overrightarrow{\mathbf{G}}_{con-acycl}^{or}(n, m)$.

Задача построения процесса $(v' \rightarrow v'')$ -pebbling ($\Gamma^{-1}(v') = \emptyset$) для орграфа $G = (V, E) \in \overrightarrow{\mathbf{G}}_{con-acycl}^{or}(n, m)$ допускает естественную декомпозицию, формулируемую следующим образом. Будем говорить, что орграф $G = (V, E) \in \overrightarrow{\mathbf{G}}_{con-acycl}^{or}(n, m)$ представим в виде последовательного соединения компонент, если G состоит из таких подграфов $G_i = (V_i, E_i)$ ($i = 1, \dots, l$), что:

- 1) $v' \in V_1$ и $v'' \in V_l$;
- 2) $V_i \cap V_{i+1} = \{v_{j_i}\}$ для всех $i = 1, \dots, l-1$;
- 3) $\{v_{j_1}, \dots, v_{j_{l-1}}\}$ – множество точек сочленения орграфа G ;
- 4) истинно равенство

$$\bigcup_{i=1}^l V_i = V,$$

причем множества

$$V_1 \setminus \{v_{j_1}\}, V_2 \setminus \{v_{j_1}, v_{j_2}\}, \dots, V_{l-1} \setminus \{v_{j_{l-2}}, v_{j_{l-1}}\}, V_l \setminus \{v_{j_{l-1}}\}$$

попарно не пересекаются.

Структура орграфа $G \in \overrightarrow{\mathbf{G}}_{con-acycl}^{or}(n, m)$, представимого в виде последовательного соединения компонент показана на рис. 3.4.

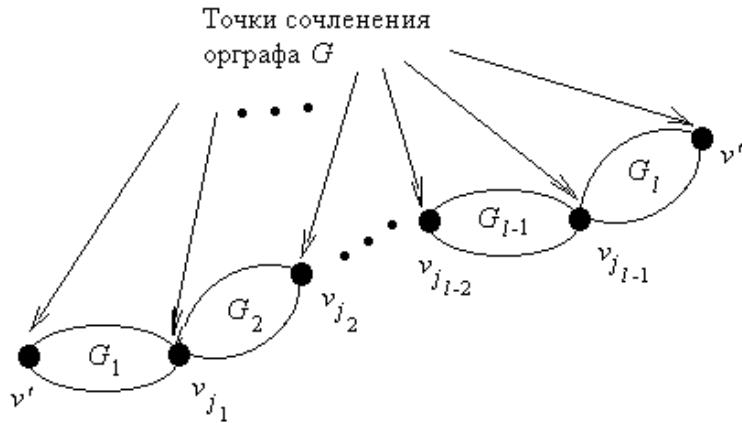


Рис. 3.4. Орграф, представимый в виде последовательного соединения компонент.

Очевидно, что для орграфа $G \in \overrightarrow{\mathbf{G}}_{con-acycl}^{or}(n, m)$, представимого в виде последовательного соединения компонент (см. рис. 3.4), задача построения процесса $(v' \rightarrow v'')$ -*pebbling* ($\Gamma^{-1}(v') = \emptyset$) сводится к независимому построению следующей l -элементной последовательности *pebbling*-процессов для компонент:

$(v' \rightarrow v_{j_1})$ -*pebbling* компоненты G_1 ,

$(v_{j_1} \rightarrow v_{j_2})$ -*pebbling* компоненты G_2 ,

.....

$(v_{j_{l-2}} \rightarrow v_{j_{l-1}})$ -*pebbling* компоненты G_{l-1} ,

$(v_{j_{l-1}} \rightarrow v'')$ -*pebbling* компоненты G_l .

Независимое построение *pebbling*-процессов для компонент дает конструктивные средства для организации *параллельных вычислений* при построении процесса $(v' \rightarrow v'')$ -*pebbling* ($\Gamma^{-1}(v') = \emptyset$) для орграфа $G \in \overrightarrow{\mathbf{G}}_{con-acycl}^{or}(n, m)$. При этом, *степень параллелизма* равна числу l выделенных компонент орграфа $G \in \overrightarrow{\mathbf{G}}_{con-acycl}^{or}(n, m)$. Более того, сложность процесса $(v' \rightarrow v'')$ -*pebbling* ($\Gamma^{-1}(v') = \emptyset$) для орграфа $G \in \overrightarrow{\mathbf{G}}_{con-acycl}^{or}(n, m)$ сле-

дующим образом выражается через сложности *pebbling*-процессов компонент. Пусть k_1 и k_l – минимальное число камешков, необходимых, соответственно, для процесса $(v' \rightarrow v_{j_1})$ -*pebbling* компоненты G_1 и для процесса $(v_{j_{l-1}} \rightarrow v'')$ -*pebbling* компоненты G_l , а k_i ($i = 2, \dots, l-1$) – минимальное число камешков, необходимых для $(v_{j_{i-1}} \rightarrow v_{j_i})$ -*pebbling* компоненты G_i . Тогда минимальное число k_G камешков, необходимых для процесса $(v' \rightarrow v'')$ -*pebbling* орграфа G вычисляется следующим образом

$$k_G = \max\{k_1, k_2, \dots, k_l\}.$$

Учитывая возможность декомпозиции построения *pebbling*-процесса, без ограничения общности можно считать, что рассматриваемый орграф $G \in \vec{\mathbf{G}}^{con-acycl}(n, m)$ не имеет точек сочленения.

Рассмотрим модельную задачу построения *pebbling*-процесса для специального типа орграфов, представляющих собой h -мерный куб ($h \in \mathbf{N}$), имеющий n^h ($n \in \mathbf{N}$) вершин. Формально, такой орграф имеет следующий вид:

$$\mathbf{cube}_h(n) = (\mathbf{V}, \mathbf{E}), \quad (n, h \in \mathbf{N})$$

где

$$\mathbf{V} = \underbrace{\mathbf{N}_n \times \dots \times \mathbf{N}_n}_{h \text{ раз}} \quad (\mathbf{N}_n = \{1, 2, \dots, n\}),$$

а множество дуг

$$\mathbf{E} \subset (\mathbf{V} \times \mathbf{V}) \setminus \{(\mathbf{v}, \mathbf{v}) \mid \mathbf{v} \in \mathbf{V}\}$$

определяется следующим образом: если

$$\mathbf{v} = (v_1, \dots, v_h) \in \mathbf{V}, \quad \mathbf{u} = (u_1, \dots, u_h) \in \mathbf{V},$$

то $(\mathbf{v}, \mathbf{u}) \in \mathbf{E}$ тогда и только тогда, когда выполнены следующие два условия:

- 1) $(\forall i \in \{1, \dots, h\})(v_i \leq u_i)$;
- 2) $\sum_{i=1}^h |u_i - v_i| = 1$.

Зафиксируем вершины $\mathbf{1} = (\underbrace{1, \dots, 1}_{h \text{ раз}}) \in \mathbf{E}$, $\mathbf{n} = (\underbrace{n, \dots, n}_{h \text{ раз}}) \in \mathbf{E}$ и оценим

сложность процесса $(\mathbf{1} \rightarrow \mathbf{n})$ -*pebbling* h -мерного куба $\mathbf{cube}_h(n) = (\mathbf{V}, \mathbf{E})$ ($n, h \in \mathbf{N}$).

Обозначим через $k_{\text{cube}}(n, h)$ ($n, h \in \mathbf{N}$) минимальное число камешков, необходимых для того, чтобы осуществить процесс $(\mathbf{1} \rightarrow \mathbf{n})$ -pebbling h -мерного куба $\text{cube}_h(n) = (\mathbf{V}, \mathbf{E})$.

Если $n = 1$, то орграф $\text{cube}_h(n)$ состоит из единственной вершины, т.е. является тривиальным. Следовательно, $k_{\text{cube}}(1, h) = 1$ для всех $h \in \mathbf{N}$. Поэтому всюду в дальнейшем предполагается, что $n > 1$.

Пример 3.2. На рис 3.5-3.7 изображены графы $\text{cube}_h(n) = (\mathbf{V}, \mathbf{E})$ ($n \in \mathbf{N}$) для случаев, когда $h \in \{1, 2, 3\}$.

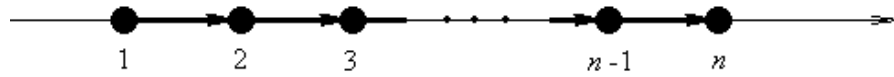


Рис. 3.5. 1-мерный куб $\text{cube}_1(n)$.

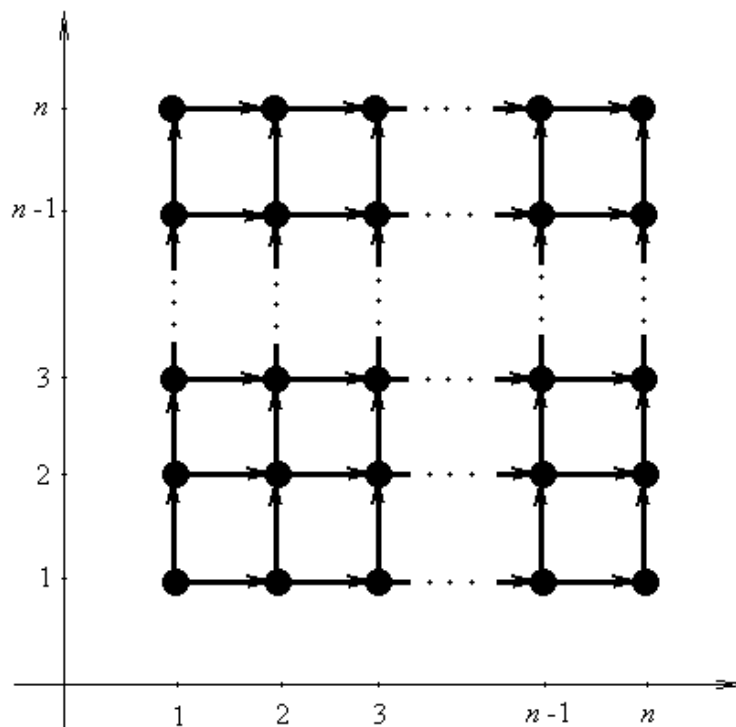


Рис. 3.6. 2-мерный куб $\text{cube}_2(n)$.

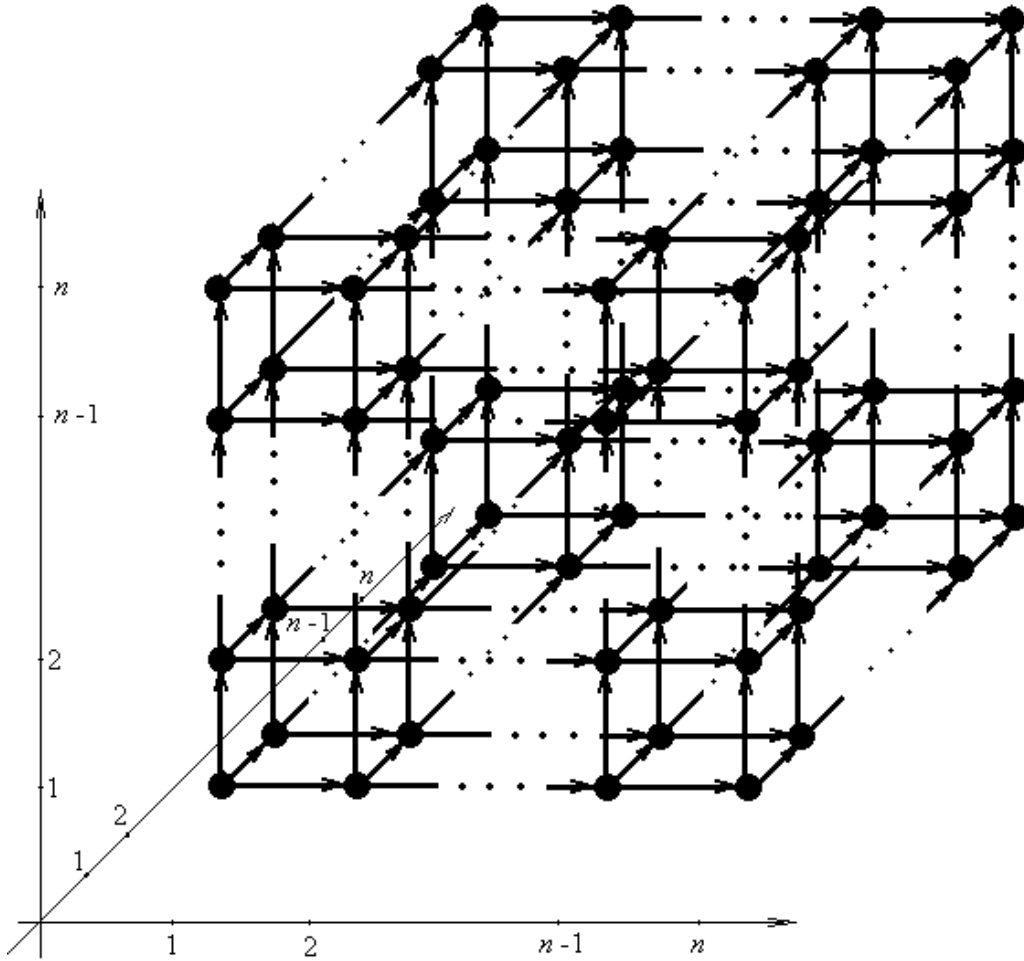


Рис. 3.7. 3-мерный куб $\mathbf{cube}_3(n)$.

Ясно, что для сложности процесса $(1 \rightarrow n)$ -pebbling 1-мерного куба $\mathbf{cube}_1(n)$ и процессов $(1 \rightarrow n)$ -pebbling 2-мерного куба $\mathbf{cube}_2(n)$ и 3-мерного куба $\mathbf{cube}_3(n)$ истинными являются следующие оценки.

$$k_{\mathbf{cube}}(n,1) = 2 \quad (n \in \mathbf{N}, n \geq 2),$$

$$k_{\mathbf{cube}}(n,2) \leq n^2 \quad (n \in \mathbf{N}, n \geq 2),$$

$$k_{\mathbf{cube}}(n,3) \leq n^3 \quad (n \in \mathbf{N}, n \geq 2).$$

Исследуем сложность процесса $(\mathbf{1} \rightarrow \mathbf{n})$ -*pebbling* h -мерного куба $\mathbf{cube}_h(n)$ ($n, h \in \mathbf{N}, n \geq 2$) в случае, когда $h \geq 2$.

Рассмотрим вначале случай, когда $h = 2$, т.е. исследуем сложность процесса $(\mathbf{1} \rightarrow \mathbf{n})$ -*pebbling* 2-мерного куба $\mathbf{cube}_2(n)$ ($n \in \mathbf{N}, n \geq 2$).

Замечание 3.10. Орграф $\mathbf{cube}_2(n)$ ($n \in \mathbf{N}, n \geq 2$) представляет собой специальный случай орграфа

$$\mathbf{rect}_{a,b} = (V, E) \quad (a, b \in \mathbf{N}; a, b \geq 2),$$

где $V = \{1, \dots, a\} \times \{1, \dots, b\}$, а множество $\mathbf{E} \subset (\mathbf{V} \times \mathbf{V}) \setminus \{(\mathbf{v}, \mathbf{v}) \mid \mathbf{v} \in \mathbf{V}\}$, определяется следующим образом: если $\mathbf{v} = (v_1, v_2) \in \mathbf{E}$ и $\mathbf{u} = (u_1, u_2) \in \mathbf{E}$, то

$$(\mathbf{v}, \mathbf{u}) \in \mathbf{E} \Leftrightarrow (v_1 \leq u_1) \& (v_2 \leq u_2) \& (|u_1 - v_1| + |u_2 - v_2| = 1).$$

Действительно, для всех $n \in \mathbf{N}$ ($n \geq 2$) истинно равенство

$$\mathbf{cube}_2(n) = \mathbf{rect}_{n,n}.$$

Орграф $\mathbf{rect}_{a,b}$ ($a, b \in \mathbf{N}; a, b \geq 2$) назовем *прямоугольником* (для определенности считаем, что прямоугольник $\mathbf{rect}_{a,b}$ содержит a вершин на горизонтальной стороне и b вершин на вертикальной стороне). Обозначим через $f(a, b)$ минимальное количество камешков, необходимых для того, чтобы осуществить процесс $((1,1) \rightarrow (a,b))$ -*pebbling* прямоугольника $\mathbf{rect}_{a,b}$. Установим некоторые свойства *pebbling*-процесса прямоугольника $\mathbf{rect}_{a,b}$ (а, следовательно, и *pebbling*-процесса орграфа $\mathbf{cube}_2(n)$).

Обозначим через $S(a, b)$ минимальное (по мощности) множество камешков, необходимых для процесса $((1,1) \rightarrow (a,b))$ -*pebbling* прямоугольника $\mathbf{rect}_{a,b}$. Тогда

$$|S(a, b)| = f(a, b).$$

Так как $\mathbf{rect}_{a,b}$ и $\mathbf{rect}_{b,a}$ – изоморфные графы, то $S(a, b) = S(b, a)$, и, следовательно, $f(a, b) = f(b, a)$.

Для орграфа $G = (V, E)$ ($E \subseteq (V \times V) \setminus \{(v, v) \mid v \in V\}$) положим:

$$\mathbf{g}(v) = \Gamma^+(v) \quad (v \in V).$$

Определим *степени* отображения \mathbf{g} обычным образом, а именно: для всех $v \in V$

$$\mathbf{g}^1(v) = \mathbf{g}(v),$$

$$\mathbf{g}^{l+1}(v) = \bigcup_{v_1 \in \mathbf{g}^l(v)} \mathbf{g}(v_1) \quad (l \in \mathbf{N}).$$

Утверждение 3.1. При любом процессе $((1,1) \rightarrow (a,b))$ -*pebbling* прямоугольника $\mathbf{rect}_{a,b}$ ($a, b \in \mathbf{N}; a, b \geq 2$) хотя бы раз покрывается камешком каждая вершина (i, j) ($1 \leq i \leq a, 1 \leq j \leq b$).

Доказательство. Возможны следующие два случая.

1. Пусть $(i, j) = (a, b)$. Тогда вершина (i, j) покрывается камешком на последнем шаге любого процесса $((1,1) \rightarrow (a,b))$ -*pebbling* прямоугольника $\mathbf{rect}_{a,b}$, что и требовалось доказать.

2. Пусть $(i, j) \neq (a, b)$. Предположим противное, т.е. что существует такой процесс $((1,1) \rightarrow (a,b))$ -*pebbling* прямоугольника $\mathbf{rect}_{a,b}$, что вершина (i, j) ни разу не покрывается камешком.

Покажем, что в этом случае ни одна из вершин, принадлежащих множеству $\mathbf{g}^{l+1}((i, j))$ ($l \in \mathbf{Z}_+$), также ни разу не покрывается камешком, принадлежащим множеству $S(a, b)$. Доказательство проведем индукцией по числу l .

Пусть $l = 0$. Из определения *pebbling*-процесса вытекает, что ни одна из вершин, принадлежащих множеству

$$\mathbf{g}((i, j)) = \mathbf{g}^{0+1}((i, j)) = \mathbf{g}^1((i, j)),$$

ни разу не покрывается камешком, что и требовалось показать.

Предположим, что утверждение справедливо для всех $l = 0, 1, \dots, h$.

Пусть $l = h + 1$. Из предположения индукции вытекает, что в процессе $((1,1) \rightarrow (a,b))$ -*pebbling* прямоугольника $\mathbf{rect}_{a,b}$ ни одна из вершин v_1 , принадлежащих множеству $\mathbf{g}^{h+1}((i, j))$, ни разу не покрывается камешком. Так как

$$\mathbf{g}^{(h+1)+1}((i, j)) = \bigcup_{v_1 \in \mathbf{g}^{h+1}((i, j))} \mathbf{g}(v_1),$$

и вершина v_1 ни разу не покрывается камешком, то из определения *pebbling*-процесса вытекает, что ни одна из вершин, принадлежащих множеству $\mathbf{g}(v_1)$, а, следовательно, и множеству $\mathbf{g}^{(h+1)+1}((i, j))$, ни разу не покрывается камешком, что и требовалось показать.

Очевидно, что $(a, b) \in \mathbf{g}^{l+1}((i, j))$, если $l = \max\{a - i, b - j\} - 1$. Поэтому, из того, что вершина (i, j) ни разу не покрывается камешком, вытекает, что в процессе $((1,1) \rightarrow (a,b))$ -*pebbling* прямоугольника $\mathbf{rect}_{a,b}$ вершина (a, b) ни разу не покрывается камешком. Полученное противоречие показывает, что предположение – ложное. Следовательно, при реализации любого процесса $((1,1) \rightarrow (a,b))$ -*pebbling* прямоугольника $\mathbf{rect}_{a,b}$ каждая вершина $(i, j) \neq (a, b)$ ($1 \leq i \leq a, 1 \leq j \leq b$) хотя бы раз покрывается некоторым камешком.

Утверждение доказано.

Найдем теперь нетривиальную верхнюю оценку сложности процесса $(\mathbf{1} \rightarrow \mathbf{n})$ -*pebbling* 2-мерного куба $\mathbf{cube}_2(n)$ ($n \in \mathbf{N}, n \geq 2$).

Лемма 3.1. Истинна следующая верхняя оценка сложности $(\mathbf{1} \rightarrow \mathbf{n})$ -*pebbling* 2-мерного куба $\mathbf{cube}_2(n)$ ($n \in \mathbf{N}, n \geq 2$)

$$k_{\mathbf{cube}}(n, 2) \leq n + 1 \quad (n \in \mathbf{N}, n \geq 2). \quad (3.49)$$

Доказательство. Рассмотрим следующий алгоритм, осуществляющий процесс $(\mathbf{1} \rightarrow \mathbf{n})$ -*pebbling* 2-мерного куба $\mathbf{cube}_2(n)$ ($n \in \mathbf{N}, n \geq 2$) (обозначим через $\mu(i, j)$ камешек, помещенный в вершину (i, j) , а через $\mu \rightarrow (i, j)$ – укладку камешка в вершину (i, j)).

Алгоритм 3.7.

```

/* инициализация значений */
begin
  do  $i = 1, \dots, n$ 
     $\mu \rightarrow (i, 1)$ 
  end
   $\mu \rightarrow (1, 2)$ 

  /* продвижение вперед */
  do  $j = 1, \dots, n - 2$ 
    do  $i = 1, \dots, n - 1$ 
       $\mu(i, j) \rightarrow (i + 1, j + 1)$ 
    end
     $\mu(n, j) \rightarrow (1, j + 1)$ 
  end

  /* последняя итерация */
  do  $i = 1, \dots, n - 1$ 
     $\mu(i, n - 1) \rightarrow (i + 1, n)$ 
  end
end

```

Корректность алгоритма 3.7 очевидна. Число камешков, которое использует этот алгоритм в процессе $(\mathbf{1} \rightarrow \mathbf{n})$ -*pebbling* 2-мерного куба $\mathbf{cube}_2(n)$ ($n \in \mathbf{N}, n \geq 2$), равно $n + 1$, откуда вытекает, что неравенство (3.50) – истинное, что и требовалось доказать.

Лемма доказана.

Замечание 3.11. Верхняя оценка, установленная в лемме 3.1, дает возможность установить аналогичную верхнюю оценку для сложности процесса $((1,1) \rightarrow (a,b))$ -*pebbling* прямоугольника $\mathbf{rect}_{a,b}$ ($a, b \in \mathbf{N}; a, b \geq 2$)

Утверждение 3.2. Для всех $a, b \in \mathbf{N}$ ($a, b \geq 2$) истинно неравенство

$$f(a, b) \leq \min\{a, b\} + 1. \quad (3.50)$$

Доказательство. Чтобы убедиться в истинности неравенства (3.50), достаточно применить для процесса $((1,1) \rightarrow (a,b))$ -*pebbling* прямоугольника $\mathbf{rect}_{a,b}$ следующие небольшие модификации алгоритма 3.7. Если $a \leq b$, то осуществляется последовательный (горизонтальная сторона за горизонтальной стороной) обход прямоугольника $\mathbf{rect}_{a,b}$ снизу вверх. Если же $a > b$, то осуществляется последовательный (вертикальная сторона за вертикальной стороной) обход прямоугольника $\mathbf{rect}_{a,b}$ слева направо.

Утверждение доказано.

Замечание 3.12. При доказательстве леммы 3.1 верхняя оценка сложности процесса $(\mathbf{1} \rightarrow \mathbf{n})$ -*pebbling* 2-мерного куба $\mathbf{cube}_2(n)$ ($n \in \mathbf{N}, n \geq 2$) установлена с помощью построения *pebbling*-процесса, сложность которого равна этой оценке.

Если необходимо найти *точную* оценку сложности *pebbling*-процесса, доказательство, обычно, разбивается на 2 этапа. На 1-м этапе устанавливается (как правило, тем же методом, что и при доказательстве леммы 3.1) верхняя, а на 2-м этапе – нижняя оценка сложности *pebbling*-процесса. Часто метод доказательства нижней оценки сложности *pebbling*-процесса состоит в том, что показывается, что данное число камешков не дает возможность покрыть целевую вершину. При этом, удобно представлять все возможные схемы покрытия вершин орграфа с помощью данного числа камешков в терминах акцептора (представленного либо в явном, либо в неявном виде).

Проиллюстрируем описанный выше подход на простом примере.

Пример 3.3. Пусть $n = 2$. Покажем, что

$$k_{\mathbf{cube}}(2,2) = 3. \quad (3.51)$$

Вначале покажем, что

$$k_{\mathbf{cube}}(2,2) \leq 3, \quad (3.52)$$

т.е. что три камешка достаточно для того, чтобы осуществить процесс $((1,1) \rightarrow (2,2))$ -*pebbling* 2-мерного куба $\mathbf{cube}_2(2)$. Эту задачу, например, решает следующая схема

$$\{(1,1)\} \Rightarrow \{(1,1), (1,2)\} \Rightarrow \{(1,1), (1,2), (2,1)\} \Rightarrow \{(1,2), (2,1), (2,2)\}.$$

Теперь покажем, что

$$k_{\mathbf{cube}}(2,2) \geq 3, \quad (3.53)$$

т.е. что два камешка недостаточно для того, чтобы осуществить процесс $((1,1) \rightarrow (2,2))$ -*pebbling* 2-мерного куба $\mathbf{cube}_2(2)$.

Предположим, что в процессе покрытия вершин 2-мерного куба $\mathbf{cube}_2(2)$ используется два камешка. Все допустимые схемы таких покрытий представлены акцептором, изображенным на рис. 3.8 (через $\mu \rightarrow (i, j)$ обозначено действие: «убрать камешек из вершины (i, j) »). Так как ни одна из схем не приводит к покрытию камеш-

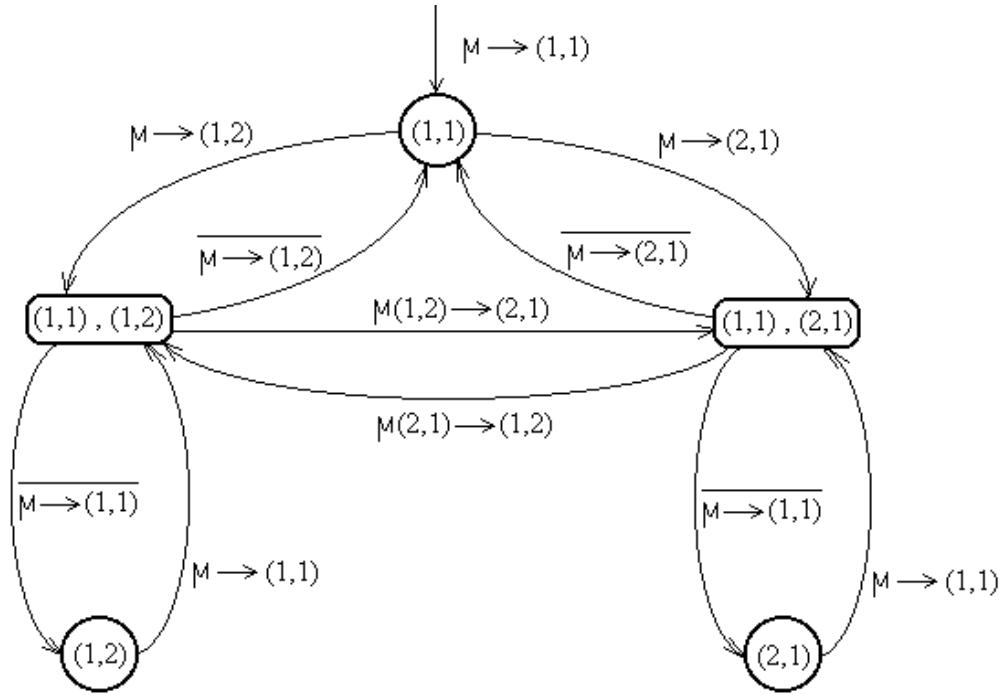


Рис. 3.8. Допустимые схемы покрытия вершин куба $\mathbf{cube}_2(2)$ двумя камешками.

ком вершины $(2,2)$, то не существует процесс $((1,1) \rightarrow (2,2))$ -pebbling 2-мерного куба $\mathbf{cube}_2(2)$ с использованием двух камешков, что и требовалось показать.

Из истинности неравенств (3.52) и (3.53) вытекает, что равенство (3.51) – истинное, что и требовалось показать.

Рассмотрим теперь случай, когда $h = 3$, т.е. исследуем сложность процесса $(1 \rightarrow n)$ -pebbling 3-мерного куба $\mathbf{cube}_3(n)$ ($n \in \mathbf{N}, n \geq 2$).

Лемма 3.2. Истинна следующая верхняя оценка сложности $(1 \rightarrow n)$ -pebbling 3-мерного куба $\mathbf{cube}_2(n)$ ($n \in \mathbf{N}, n \geq 2$)

$$k_{\mathbf{cube}}(n, 3) \leq n^2 + n \quad (n \in \mathbf{N}, n \geq 2). \quad (3.54)$$

Доказательство. Обозначим через $\mathit{pebble}(*, i_2, i_3)$ ($i_2, i_3 \in \{1, \dots, n\}$) процедуру, состоящую в последовательном (одна за другой) замощении камешками вершин 3-мерного куба $\mathbf{cube}_2(n)$ ($n \in \mathbf{N}, n \geq 2$), принадлежащих множеству

$$(*, i_2, i_3) = \{(i_1, i_2, i_3) \mid i_1 \in \{1, \dots, n\}\} \quad (i_2, i_3 \in \{1, \dots, n\}),$$

а через $move(*, i_2, i_3) \rightarrow (*, i'_2, i'_3)$ ($i_2, i'_2, i_3, i'_3 \in \{1, \dots, n\}$) – процедуру, состоящую в последовательном (один за другим) переносе камешков из множества вершин $(*, i_2, i_3)$ в множество вершин $(*, i'_2, i'_3)$ в соответствии со схемой

$$\mu(i_1, i_2, i_3) \rightarrow (i_1, i'_2, i'_3) \quad (i_1 \in \{1, \dots, n\}).$$

Рассмотрим следующий алгоритм, осуществляющий процесс $(\mathbf{1} \rightarrow \mathbf{n})$ -*pebbling* 3-мерного куба $\mathbf{cube}_3(n)$ ($n \in \mathbf{N}, n \geq 2$).

Алгоритм 3.8.

```

/* инициализация значений */
begin
  do  $i_2 = 1, \dots, n$ 
     $pebble(*, i_2, 1)$ 
  end
   $pebble(*, 1, 2)$ 

  /* продвижение вперед */
  do  $i_3 = 1, \dots, n - 2$ 
    do  $i_2 = 1, \dots, n - 1$ 
       $move(*, i_2, i_3) \rightarrow (*, i_2 + 1, i_3 + 1)$ 
    end
     $move(*, n, i_3) \rightarrow (*, 1, i_3 + 1)$ 
  end

  /* последняя итерация */
  do  $i_2 = 1, \dots, n - 1$ 
     $move(*, i_2, n - 1) \rightarrow (*, i_2 + 1, n)$ 
  end
end

```

Корректность алгоритма 3.8 очевидна. Число камешков, которое использует этот алгоритм в процессе $(\mathbf{1} \rightarrow \mathbf{n})$ -*pebbling* 3-мерного куба $\mathbf{cube}_3(n)$ ($n \in \mathbf{N}, n \geq 2$), равно $n^2 + n$, откуда вытекает, что неравенство (3.54) – истинное, что и требовалось доказать.

Лемма доказана.

Нетрудно заметить, что алгоритмы 3.7 и 3.8 имеют одну и ту же структуру. Более того, процедура *pebble* представляет собой, по сути, применение алгоритма 3.7 в алгоритме 3.8. Отсюда непосредственно вытекает следующее. Во-первых, целесообразно (а главное, не очень сложно) применить рекурсию при построении процесса $(\mathbf{1} \rightarrow \mathbf{n})$ -*pebbling* h -мерного куба $\mathbf{cube}_h(n)$ ($n, h \in \mathbf{N}; n, h \geq 2$). Во-вторых, применение рекурсии обосновывает истинность следующей асимптотической оценки сложности процесса $(\mathbf{1} \rightarrow \mathbf{n})$ -*pebbling* h -мерного куба $\mathbf{cube}_h(n)$ ($n, h \in \mathbf{N}; n, h \geq 2$).

Теорема 3.9. Истинна следующая асимптотическая оценка сложности процесса $(\mathbf{1} \rightarrow \mathbf{n})$ -*pebbling* h -мерного куба $\mathbf{cube}_h(n)$ ($n, h \in \mathbf{N}; n, h \geq 2$)

$$k_{\mathbf{cube}}(n, h) = O(n^{h-1}) \quad (n \rightarrow \infty). \quad (3.55)$$

Замечание 3.13. Орграф $\mathbf{rect}_{a,b}$ ($a, b \in \mathbf{N}; a, b \geq 2$) (см. замечания 3.10 и 3.11) представляет собой специальный случай h -мерного параллелепипеда

$$\mathbf{prlppd}_{a_1, \dots, a_h} = (V, E), \quad (a_1, \dots, a_h \in \mathbf{N}; a_1, \dots, a_h \geq 2)$$

где $V = \{1, \dots, a_1\} \times \dots \times \{1, \dots, a_h\}$, а множество $\mathbf{E} \subset (\mathbf{V} \times \mathbf{V}) \setminus \{(\mathbf{v}, \mathbf{v}) \mid \mathbf{v} \in \mathbf{V}\}$, определяется следующим образом: если $\mathbf{v} = (v_1, \dots, v_h) \in \mathbf{E}$ и $\mathbf{u} = (u_1, \dots, u_h) \in \mathbf{E}$, то

$$(\mathbf{v}, \mathbf{u}) \in \mathbf{E} \Leftrightarrow (\forall i \in \{1, \dots, h\})(v_i \leq u_i) \& \left(\sum_{i=1}^h |u_i - v_i| = 1 \right).$$

Орграф $\mathbf{cube}_h(n)$ ($n, h \in \mathbf{N}; n, h \geq 2$), в свою очередь, представляет собой специальный случай орграфа $\mathbf{prlppd}_{a_1, \dots, a_h}$ ($a_1, \dots, a_h \in \mathbf{N}; a_1, \dots, a_h \geq 2$), что непосредственно вытекает из истинности равенства

$$\mathbf{cube}_h(n) = \mathbf{prlppd}_{\underbrace{n, \dots, n}_{h \text{ раз}}}.$$

Обозначим через $f(a_1, \dots, a_h)$ минимальное количество камешков, необходимых для того, чтобы осуществить процесс $((\underbrace{1, \dots, 1}_{h \text{ раз}}) \rightarrow (a_1, \dots, a_h))$ *pebbling* орграфа

$\mathbf{prlppd}_{a_1, \dots, a_h}$ ($a_1, \dots, a_h \in \mathbf{N}; a_1, \dots, a_h \geq 2$). Из (3.55) вытекает истинность следующей асимптотической оценки

$$f(a_1, \dots, a_h) = O((\max\{a_i \mid i = 1, \dots, h\})^{h-1}) \quad (a_1, \dots, a_h \rightarrow \infty).$$

В заключение отметим, что утверждение 3.1 легко обобщается на случай орграфа $\mathbf{prlppd}_{a_1, \dots, a_h}$ ($a_1, \dots, a_h \in \mathbf{N}; a_1, \dots, a_h \geq 2$).

3.6. Выводы.

Полученные в настоящем разделе результаты показывают, что для основных задач теории графов, связанных с построением путей, циклов и остовных деревьев типичной является следующая ситуация. Задача построения одного (безразлично, какого именно) объекта принадлежит классу $Lcl(\mathbf{R})$, где \mathbf{R} – представление графов и/или орграфов, либо списками, либо матрицами смежности (либо вершин, либо ребер). В то же время, число конструируемых объектов является экспонентой от числа вершин исследуемого графа (ряд таких нетривиальных оценок для числа конструируемых объектов установлен в настоящем разделе). Отсюда непосредственно вытекает, что задачи построения всех объектов данного типа заведомо не принадлежат классу $Lcl(\mathbf{R})$ при любом представлении \mathbf{R} графа и/или орграфа. Однако, как показано в настоящем разделе, эти задачи принадлежат классу $WORK - Lcl(\mathbf{R})$, где \mathbf{R} – представление графов, либо списками, либо матрицами смежности (либо вершин, либо ребер).

Задачи, связанные с перемещениями по графу и/или орграфу в соответствии с определенными правилами к заданной цели имеют многочисленные применения, как в фундаментальных, так и прикладных исследованиях. С этой точки зрения особый интерес представляет *pebbling*, сфера приложений которого определяется многочисленными применениями методов сетевого планирования и управления. Принадлежность задачи построения одного (безразлично, какого именно) *pebbling*-процесса классу $Lcl(\mathbf{R})$, где \mathbf{R} – представление орграфов, либо списками, либо матрицами смежности (либо вершин, либо ребер), гарантирует эффективность его применения при разработке алгоритмов на графах. Из установленной в настоящем разделе асимптотической оценки сложности процесса $(\mathbf{1} \rightarrow \mathbf{n})$ -*pebbling* h -мерного куба $\mathbf{cube}_h(n)$ ($n, h \in \mathbf{N}; n, h \geq 2$) вытекает достаточно низкая *относительная сложность* этого процесса, т.е. число используемых камешков, приходящихся на одну вершину. Действительно, из (3.55) вытекает, что асимптотическая оценка *относительной сложности* процесса $(\mathbf{1} \rightarrow \mathbf{n})$ -*pebbling* h -мерного куба $\mathbf{cube}_h(n)$ ($n, h \in \mathbf{N}; n, h \geq 2$) равна $O(n^{-1})$ ($n \rightarrow \infty$), т.е. вообще не зависит от числа h .

В заключение отметим, что, по-видимому, перспективным является исследование задач построения *pebbling*-процессов для специальных, достаточно узких, классов связанных ациклических орграфов, в том числе описание классов таких подграфов, для которых существует эффективная декомпозиция задач построения *pebbling*-процессов.

4. ЗА ПРЕДЕЛАМИ КЛАССА $WORK - Lcl(\mathbf{R})$.

Классы $Lcl(\mathbf{R})$ и $WORK - Lcl(\mathbf{R})$ состоят из задач теории графов, разрешимых с помощью алгоритмов, *простейших* с точки зрения их емкостной сложности. Этим классам задач заведомо не принадлежат задачи построения объектов, чья емкостная сложность – экспонента от числа вершин исходных графов, причем в принципе невозможна последовательная генерация (один за другим) фрагментов этих объектов в рамках линейной памяти, до полного их построения. Основная цель настоящего раздела состоит в том, чтобы показать, что существует ряд модельных задач, как дискретной математики, так и теории графов, которые не принадлежат классу $WORK - Lcl(\mathbf{R})$.

Структура настоящего раздела – следующая. В п.4.1 сделан ряд замечаний, в частности, связанных с техникой, используемой в настоящем разделе. В п.4.2 показано, что классу $WORK - Lcl(\mathbf{R})$ не принадлежат задачи, как идентификации начального состояния, так и синхронизации множества допустимых начальных состояний, для конечного слабоинициального автомата – модельные задачи теории автоматов, естественно сводящиеся к построению специальных стратегий прогулок по направленному мультиграфу (возможно, с петлями) с раскрашенными дугами. В п.4.3 показано, что классу $WORK - Lcl(\mathbf{R})$ не принадлежит задача построения супервизора для адаптивного управления системой дискретных событий, представленной моделью автоматного типа. Эта задача сводится к построению специальной адаптивной стратегии прогулки (т.е. прогулки, осуществляемой конечным автоматом) по направленному мультиграфу (возможно, с петлями) с раскрашенными дугами. В п.4.4 показано, что классу $WORK - Lcl(\mathbf{R})$ не принадлежит задачи построения выигрышных стратегий для игр двух лиц на графе, состоящих в построении пути между двумя вершинами. П.4.5 содержит краткие выводы.

4.1. Предварительные замечания.

В п.3.1 было отмечено, что поиск является универсальным и, часто, единственным методом решения достаточно широкого класса задач дискретной математики и ее многочисленных приложений. Основная причина такого положения состоит в следующем. Поиск, в качестве основной операции, фактически заложен, в явном или неявном виде, во всех остальных действиях с рассматриваемыми теоретико-множественными (часто, по своей сути, лингвистическими) объектами дискретной математики. Кроме того, для действий с этими объектами (именно в силу их теоретико-множественной природы), в принципе отсутствует эффективный алгебраический аппарат, предназначенный именно для вычислений, а не для дескриптивных описаний в терминах групп эндоморфизмов или автоморфизмов (которые сами по себе могут быть объектами с достаточно сложной структурой). В таких случаях решение задач дискретной математики методом поиска (т.е. построение последовательности (3.1)), по своей сути, сводится к построению и просеиванию с помощью правил отсекающего достаточно большого по числу элементов множества достаточно длинных последовательностей (теоретико-множественных) объектов вида

$$a_{kj} = F_j(a_{1j_1} \dots a_{k-1,j_{k-1}}) \quad (k = 1, 2, \dots; F_j \in F),$$

где F – конечное множество порождающих элементов полугруппы операторов. Ясно, что в большинстве случаев неограниченный рост, как длины, так и числа конструируемых и анализируемых последовательностей объектов, приводит к экспоненциальной емкостной сложности поиска. Поэтому все задачи дискретной математики и ее приложений, принадлежащие к указанному типу задач, которые естественно сводятся к задаче на графе (направленном графе, мультиграфе и т.д.), заведомо порождают задачи теории графов, не принадлежащие классу $WORK - Lcl(\mathbf{R})$.

В ряде случаев построение указанных выше последовательностей элементов $\{a_{kj}\}$ естественно сводится к построению стратегии прогулки специального вида по некоторому графу. Таким образом, за пределы класса $WORK - Lcl(\mathbf{R})$ заведомо выводят следующие задачи дискретной математики и ее приложений.

Во-первых, это задачи, сводящиеся к построению стратегий не представимых в рамках линейной памяти таких прогулок по графу, что в принципе невозможна последовательная генерация (один за другим) фрагментов этих прогулок в рамках линейной памяти, до полного их построения.

Во-вторых, это ряд задач, сводящихся к построению сложных адаптивных стратегий прогулок по графу. Такие стратегии допускают естественное представление в терминах специальных конечных автоматов (или их коллективов) с экспоненциальным от числа вершин графа числом состояний, осуществляющих специальный обход графа. Этот класс задач имеет непосредственную связь с задачами исследования поведения конечных автоматов в (необязательно прямоугольных) лабиринтах.

В третьих, это ряд задач, сводящихся к построению выигрышных стратегий (которые также допускают естественное представление в терминах специальных конечных автоматов) в игре двух лиц на графе. Здесь можно выделить следующие два типа игр на графе. К первому типу игр относятся игры, цель которых состоит в построении определяемого в терминах исходного графа объекта, не представимого в рамках линейной памяти и не допускающего построения в виде фрагментов, представимых в рамках линейной памяти. Ко второму типу игр относятся игры, цель которых состоит в построении представимого в рамках линейной памяти объекта, определяемого в терминах исходного графа, но вычисление характеристической функции множества партий, приводящих к выигрышу, эквивалентно решению ряда задач типа БФК (*БУЛЕВСКИЕ ФОРМУЛЫ С КВАНТОРАМИ*).

Последняя задача представляет собой классическую $P-SPACE-COMplete$ задачу, формулируемую, обычно, следующим образом (см., напр. [3])

БУЛЕВСКИЕ ФОРМУЛЫ С КВАНТОРАМИ (БФК)

Условие: Задана правильно построенная булевская формула

$$F = (Q_1 x_1)(Q_2 x_2) \dots (Q_n x_n) f(x_1, \dots, x_n),$$

где $Q_i \in \{\forall, \exists\}$ ($i = 1, \dots, n$), а $f(x_1, \dots, x_n)$ – булевское выражение с переменными x_1, \dots, x_n .

Вопрос: Верно ли, что формула F истинна?

Отметим, что сложность решения задачи БФК определяется именно формулами с чередующейся последовательностью кванторов \forall и \exists – типичными характеристическими функциями при построении выигрышных стратегий в играх двух лиц.

4.2. Задачи идентификации состояний автоматов.

Пусть $M = (Q, X, Y, \delta, \lambda)$ – заданный автомат (Мили) (Q – множество состояний, X и Y – соответственно, входной и выходной алфавит, $\delta : Q \times X \rightarrow Q$ и $\lambda : Q \times X \rightarrow Y$ – функция, соответственно, переходов и выходов). Функции переходов и выходов автомата M , как правило, расширяются до функций $\tilde{\delta} : Q \times X^* \rightarrow Q$ и $\tilde{\lambda} : Q \times X^* \rightarrow Y$ с помощью равенств:

$$\tilde{\delta}(q, \Lambda) = q, \quad \tilde{\delta}(q, px) = \delta(\tilde{\delta}(q, p), x),$$

$$\tilde{\lambda}(q, \Lambda) = q, \quad \tilde{\lambda}(q, px) = \tilde{\lambda}(q, p)\lambda(\tilde{\delta}(q, p), x),$$

Зафиксируем множество Q_0 ($Q_0 \subseteq Q, |Q_0| \geq 2$) допустимых начальных состояний. Пара (M, Q_0) называется слабоинициальным автоматом. Для слабоинициального автомата (M, Q_0) входное слово $p \in X^+$ называется:

1) *диагностическим*, если

$$(\forall q', q'' \in Q_0)(\tilde{\lambda}(q', p) = \tilde{\lambda}(q'', p) \Rightarrow q' = q'');$$

2) *синхронизирующим*, если

$$(\forall q', q'' \in Q_0)(\tilde{\delta}(q', x) = \tilde{\delta}(q'', x));$$

3) *установочным*, если

$$(\forall q', q'' \in Q_0)(\tilde{\lambda}(q', p) = \tilde{\lambda}(q'', p) \Rightarrow \tilde{\delta}(q', p) = \tilde{\delta}(q'', p)).$$

Из этих определений вытекает, что:

1) диагностическое слово идентифицирует *начальное* состояние заданного слабоинициального автомата (т.е. то состояние, в котором автомат находился до подачи на него входного слова);

2) установочное слово идентифицирует *финальное* состояние заданного слабоинициального автомата (т.е. то состояние, в которое автомат перейдет после подачи на него слова);

3) синхронизирующее слово идентифицирует *финальное* состояние заданного слабоинициального автомата за счет *стягивания* множества Q_0 в одноэлементное множество.

По этой причине диагностические, синхронизирующие и установочные слова для слабоинициального автомата (M, Q_0) называются *идентифицирующими* словами. Известно, что задачи построения минимальных (по длине) идентифицирующих слов для заданного слабоинициального автомата являются модельными задачами теории автоматов. Поэтому сложность решения этих задач, во многом, определяет сложность решения для широкого класса задач теории автоматов.

Замечание 4.1. Естественным обобщением идентифицирующих слов, имеющим многочисленные приложения на практике, являются слова, идентифицирующие состояние автомата с точностью до класса некоторого отношения эквивалентности, определенного на множестве состояний. Разработкой методов и исследованием сложности построения таких слов занимались многие авторы. Чтобы не усложнять изложение, ограничимся рассмотрением только тех слов, которые идентифицируют состояние слабоинициального автомата в обычном смысле, т.е. так, как это было определено выше.

Обозначим через $L^d(M, Q_0)$, $L^h(M, Q_0)$ и $L^s(M, Q_0)$ минимальную длину, соответственно, диагностического, установочного и возвратного слова для слабоинициального автомата (M, Q_0) , а через $\mathbf{M}_{k,m,n}$ – множество всех автоматов $M = (Q, X, Y, \delta, \lambda)$, имеющих фиксированные множества состояний $Q = \{1, \dots, k\}$, входной алфавит $X = \{1, \dots, m\}$ и выходной алфавит $Y = \{1, \dots, n\}$. Определим отображения

$$L_{k,m,n}^u : \{2, \dots, k\} \rightarrow \mathbf{Z}_+ \quad (u \in \{d, h, s\})$$

равенством

$$L_{k,m,n}^u(r) = \max L_{k,m,n}^u(M, Q_0) \quad (r \in \{2, \dots, k\}),$$

где максимум берется по всем $M \in \mathbf{M}_{k,m,n}$ и всем таким $Q_0 \subseteq Q$, что $|Q_0| = r$ и положим

$$L_{k,m,n}^u = \max_{r \in \{2, \dots, k\}} L_{k,m,n}^u(r).$$

Таким образом, $L_{k,m,n}^d(r)$, $L_{k,m,n}^h(r)$, $L_{k,m,n}^s(r)$, $L_{k,m,n}^d$, $L_{k,m,n}^h$ и $L_{k,m,n}^s$ представляют собой *функции Шеннона*, характеризующие сложность построения минимальных идентифицирующих слов. Исследованию этих функций Шеннона посвящен ряд работ. В 1965 году *T.N. Hibbard* установил следующую точную оценку максимальной длины минимального установочного слова

$$L_{k,m,n}^h(r) = 0.5 \cdot (2 \cdot k - r) \cdot (r - 1) \quad (r \in \{2, \dots, k\}).$$

В 1970 году *Z. Kohavi* доказал, что в специальном случае, когда $Q_0 = Q$, истинной является следующая верхняя оценка длины минимального синхронизирующего слова

$$L_{k,m,n}^s(k) \leq 0.5 \cdot k \cdot (k - 1)^2.$$

Имеется много невыясненных моментов, связанных с оценками длин минимальных диагностических и синхронизирующих слов. Верхние оценки

$$L_{k,m,n}^d(r) \leq (r - 1) \cdot k^r \quad (r \in \{2, \dots, k\}),$$

$$L_{k,m,n}^d(r) \leq k! \quad (r \in \{2, \dots, k\})$$

установили, соответственно, *A. Gill* в 1965 году и *P. Starke* в 1967 году.

В 1976 году *М.Н. Соколовский* показал, что истинными являются верхняя оценка

$$L_{k,m,n}^d(r) \leq \begin{cases} (r - 1) \cdot k^{0.5 \cdot k \cdot (1 + \varepsilon)}, & \text{если } r \in \{2, \dots, k - 1\} \\ \binom{k}{0.5 \cdot k} \cdot k^2, & \text{если } r = k \end{cases},$$

где $\varepsilon \rightarrow 0$, если $r \rightarrow \infty$, а также нижняя оценка

$$L_{k,m,n}^d(r) \geq \begin{cases} \binom{k-1}{r-1}, & \text{если } r \in \{2, \dots, \lfloor 0.5 \cdot k \rfloor\} \\ \binom{k-2}{\lfloor 0.5 \cdot (k-2) \rfloor}, & \text{если } r \in \{\lfloor 0.5 \cdot k \rfloor + 1, \dots, k - 1\}. \\ 3^{\lfloor \frac{1}{6} \cdot k \rfloor}, & \text{если } r = k \end{cases}$$

В 1978 году *И.К. Рыцков* доказал, что в специальном случае, когда $Q_0 = Q$, истинной является следующая точная асимптотическая оценка

$$\log_3 L_{k,m,n}^d(k) \sim \frac{k}{6} \quad (k \rightarrow \infty).$$

Работы *М.Н. Соколовского* и *И.К. Рыцова* объединяет то, что используемые в них автоматы имеют входной алфавит, мощность которого, фактически, совпадает с длиной минимального диагностического слова. Однако в 1987 году *В.Г. Скобелев* показал, что уже для случая автомата с двухбуквенным входным алфавитом истинными являются следующие экспоненциальные нижние оценки для функций Шеннона $L_{k,m,n}^d$ и $L_{k,m,n}^s$

$$L_{k,2,n}^d \geq e^{O(\sqrt{k})} \quad (k \rightarrow \infty),$$

$$L_{k,2,n}^s \geq e^{O(\sqrt{k})} \quad (k \rightarrow \infty).$$

Автомат $M \in \mathbf{M}_{k,m,n}$ представляет собой модель нечисловой природы. Как следствие, единственно возможным методом построения минимальных идентифицирующих слов для слабоинициального автомата (M, Q_0) ($M \in \mathbf{M}_{k,m,n}$) является *поиск*. Следовательно, сложность построения минимальных идентифицирующих слов для слабоинициального автомата (M, Q_0) ($M \in \mathbf{M}_{k,m,n}$) определяется сложностью соответствующего поиска. Очевидно, что построение минимальных установочных слов для слабоинициального автомата (M, Q_0) ($M \in \mathbf{M}_{k,m,n}$) осуществимо в рамках памяти $O(k^2)$ ($k \rightarrow \infty$). Иная ситуация имеет место при построении минимальных диагностических и синхронизирующих слов для слабоинициального автомата (M, Q_0) ($M \in \mathbf{M}_{k,m,n}$). Во-первых, как было отмечено выше, нижние оценки функций Шеннона для длин минимальных диагностических и синхронизирующих слов являются экспонентами от числа состояний автомата. Во-вторых, несложно построить примеры последовательностей слабоинициальных автоматов, для которых как длина, так и ширина дерева поиска минимальных диагностических и/или синхронизирующих слов являются функциями вида $O(k^{f(k)})$ ($k \rightarrow \infty$), где $f(k) \rightarrow \infty$, если $k \rightarrow \infty$.

Одним из способов задания автомата $M = (Q, X, Y, \delta, \lambda)$ является его представление направленным мультиграфом (возможно, с петлями) G_M , дуги которого раскрашены элементами множества $X \times Y$ в соответствии со специальными правилами. Положим

$$(X \times Y)(q, q') = \{(x, y) \in X \times Y \mid (\delta(q, x) = q') \& (\lambda(q, x) = y)\},$$

$$\text{wght}(q, q') = |\{x \in X \mid \delta(q, x) = q'\}| \quad (q, q' \in Q).$$

Направленный мультиграф G_M характеризуется следующим образом:

1) множеством вершин направленного мультиграфа G_M является множество Q ;

2) множество дуг направленного мультиграфа G_M содержит $|Q| \cdot |X|$ элементов, определяемых следующим образом: число экземпляров дуги (q, q') ($q, q' \in Q$) равно $wght(q, q')$ (равенство $wght(q, q') = 0$ означает, что дуга (q, q') вообще отсутствует), причем эти экземпляры окрашены различными элементами множества $(X \times Y)(q, q')$.

Пример 4.1. На рис. 4.1.а) автомат $M_1 = (Q, X, Y, \delta, \lambda)$ представлен в виде *автоматной таблицы* (или, иными словами, в виде *таблицы переходов-выходов*). Для этого автомата $Q = \{1, 2, 3, 4, 5, 6\}$, $X = \{0, 1\}$ и $Y = \{0, 1, 2\}$.

Направленный мультиграф G_{M_1} , представляющий автомат M_1 , изображен на рис. 4.1.б). Число его вершин равно $|Q| = 6$, а число дуг равно

$$|Q| \cdot |X| = 6 \cdot 2 = 12.$$

Для упрощения изображения обычно принимается соглашение о том, что все экземпляры дуги (q, q') представляются одним экземпляром. На этом экземпляре записываются все элементы множества $(X \times Y)(q, q')$, разделенные знаками дизъюнкции. При этом элемент $(x, y) \in (X \times Y)(q, q')$ принято записывать в виде x / y .

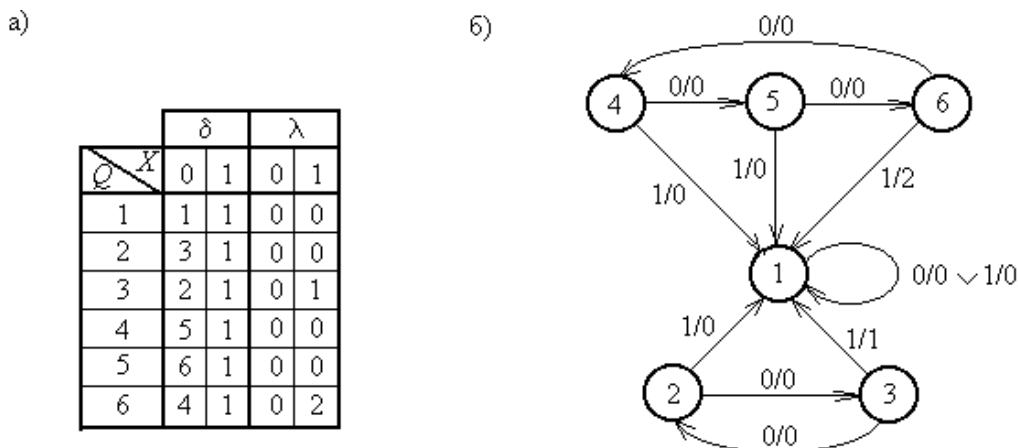


Рис. 4.1. Представление автомата M_1 : а) автоматной таблицей, б) направленным мультиграфом G_{M_1} .

Покажем, что задачи построения идентифицирующих слов для слабоинициального автомата (M, Q_0) ($M \in \mathbf{M}_{k,m,n}$) допускают естественную интерпретацию в терминах задач построения специальных *стратегий* прогулок по направленному мультиграфу G_M .

Покажем вначале, что любое входное слово

$$p = x_1 x_2 \dots x_l \in X^+$$

можно рассматривать как *стратегию* прогулки (длины l) по направленному мультиграфу G_M .

Действительно, выберем в направленном мультиграфе G_M произвольную вершину, которую обозначим через q_Λ . Существует единственная дуга $\overrightarrow{e_1}$ с началом в вершине q_Λ , окрашенная таким элементом $(x, y) \in X \times Y$, что $x = x_1$. Пусть дуга $\overrightarrow{e_1}$ заканчивается в вершине q_{x_1} . Существует единственная дуга $\overrightarrow{e_2}$ с началом в вершине q_{x_1} , окрашенная таким элементом $(x, y) \in X \times Y$, что $x = x_2$. Пусть дуга $\overrightarrow{e_2}$ заканчивается в вершине $q_{x_1 x_2}$ и т.д. В результате описанных действий для каждой вершины q_Λ направленного мультиграфа G_M единственным образом определяется следующая прогулка длины l с началом в вершине q_Λ

$$\pi_{q_\Lambda} = q_\Lambda, \overrightarrow{e_1}, q_{x_1}, \overrightarrow{e_2}, q_{x_1 x_2}, \dots, \overrightarrow{e_{l-1}}, q_{x_1 \dots x_{l-1}}, \overrightarrow{e_l}, q_{x_1 \dots x_l}.$$

Пример 4.2. Рассмотрим стратегию прогулки по направленному мультиграфу G_{M_1} (см. пример 4.1), определяемую входным словом $0^3 1^2$. В зависимости от выбора начальной вершины в направленном мультиграфе G_{M_1} , эта стратегия определяет следующие прогулки (через $(1,1)_0$ и $(1,1)_1$ обозначена петля в вершине 1, окрашенная, соответственно, элементом $(0,0) \in X \times Y$ и элементом $(1,0) \in X \times Y$):

начальная вершина – 1: $\pi_1 = 1, (1,1)_0, 1, (1,1)_0, 1, (1,1)_0, 1, (1,1)_1, 1, (1,1)_1, 1$;

начальная вершина – 2: $\pi_2 = 2, (2,3), 3, (3,2), 2, (2,3), 3, (3,1), 1, (1,1)_1, 1$;

начальная вершина – 3: $\pi_3 = 3, (3,2), 2, (2,3), 3, (3,2), 2, (2,1), 1, (1,1)_1, 1$;

начальная вершина – 4: $\pi_4 = 4, (4,5), 5, (5,6), 6, (6,4), 4, (4,1), 1, (1,1)_1, 1$;

начальная вершина – 5: $\pi_5 = 5, (5,6), 6, (6,4), 4, (4,5), 5, (5,1), 1, (1,1)_1, 1$;

начальная вершина – 6: $\pi_6 = 6, (6,4), 4, (4,5), 5, (5,6), 6, (6,1), 1, (1,1)_1, 1$.

Характеристикой прогулки

$$\pi = q_0, \overrightarrow{e_1}, q_1, \overrightarrow{e_2}, q_2, \dots, q_{l-1}, \overrightarrow{e_l}, q_l$$

по направленному мультиграфу G_M ($M \in \mathbf{M}_{k,m,n}$) назовем такое выходное слово

$$y_1 y_2 \dots y_l \in Y^*,$$

что $(x_i, y_i) \in X \times Y$ ($i = 1, \dots, l$) – окраска дуги $\overrightarrow{e_i}$.

Пример 4.3. Характеристиками прогулок

$$\pi_1, \dots, \pi_6$$

из примера 4.2 являются, соответственно, выходные слова

$$0^5, 0^3 10, 0^5, 0^5, 0^5, 0^3 20.$$

Задачи построения идентифицирующих слов для слабоинициального автомата (M, Q_0) ($M \in \mathbf{M}_{k,m,n}$) могут быть следующим образом охарактеризованы в терминах задач построения стратегий прогулок по направленному мультиграфу G_M , изображения отметок вершин которого *покрыты кляксами* (или, иными словами, *не наблюдаемы*):

1) задача построения диагностического слова для слабоинициального автомата (M, Q_0) ($M \in \mathbf{M}_{k,m,n}$) эквивалентна задаче построения такой стратегии прогулки $p \in X^+$ по направленному мультиграфу G_M , что для каждой начальной вершины $q \in Q_0$, осуществив прогулку π_q в соответствии со стратегией p , можно идентифицировать начальную вершину q на основе анализа только характеристики прогулки π_q ;

2) задача построения установочного слова для слабоинициального автомата (M, Q_0) ($M \in \mathbf{M}_{k,m,n}$) эквивалентна задаче построения такой стратегии прогулки $p \in X^+$ по направленному мультиграфу G_M , что для каждой начальной вершины $q \in Q_0$, осуществив прогулку π_q в соответствии со стратегией p , можно идентифицировать финальную вершину на основе анализа только характеристики прогулки π_q ;

3) задача построения синхронизирующего слова для слабоинициального автомата (M, Q_0) ($M \in \mathbf{M}_{k,m,n}$) эквивалентна задаче построения такой стратегии прогулки $p \in X^+$ по направленному мультиграфу G_M , что для всех начальных вершин $q \in Q_0$ завершение прогулки π_q , осуществляемой в соответствии со стратегией p , происходит в одной и той же вершине.

Пример 4.4. Рассмотрим направленный мультиграф G_{M_1} (см. пример 4.1). Положим $Q_0 = \{1, 2, 6\}$.

Стратегия прогулки по направленному мультиграфу G_{M_1} , определяемая входным словом $0^3 1^2$ (см. пример 4.1), для всех начальных вершин $q \in Q_0$ дает возможность идентифицировать как начальную, так и финальную вершины на основе анализа только характеристики соответствующей прогулки. Более того, для всех начальных вершин $q \in Q_0$, завершение прогулки, осуществляемой в соответствии с этой стратегией, происходит в одной и той же вершине, а именно, в вершине 1.

Описанные выше стратегии прогулок по направленному мультиграфу G_M ($M \in \mathbf{M}_{k,m,n}$), изображения отметок вершин которого покрыты кляксами, назовем *стратегиями идентификации множества вершин Q_0* . Выделим следующие основные типы таких стратегий.

Стратегию $p \in X^+$ идентификации множества вершин Q_0 в направленном мультиграфе G_M назовем:

1) *оптимальной*, если входное слово p – минимальное идентифицирующее слово для слабоинициального автомата (M, Q_0) ;

2) *не избыточной*, если ни одно из слов, полученных в результате вычеркивания в слове p хотя бы одной буквы, не является идентифицирующим словом для слабоинициального автомата (M, Q_0) .

Рассмотрим задачи построения стратегий идентификации множества вершин Q_0 в направленном мультиграфе G_M ($M \in \mathbf{M}_{k,m,n}$), предназначенных для идентификации финальной вершины.

Очевидно, что задача построения одной (безразлично, какой именно) оптимальной стратегии принадлежит классу $LCl(\mathbf{R})$ в следующих случаях.

Во-первых, когда \mathbf{R} – представление G_M матрицей смежности вершин.

Во-вторых, когда

$$m = \Omega(k) \quad (k \rightarrow \infty)$$

и \mathbf{R} – представление G_M либо матрицей смежности дуг, либо списками смежности (либо вершин, либо дуг).

Однако, если

$$m = o(k) \quad (k \rightarrow \infty)$$

и \mathbf{R} – представление G_M либо матрицей смежности дуг, либо списками смежности (либо вершин, либо дуг), то остается открытым даже вопрос о принадлежности рассматриваемой задачи классу $WORK - LCl(\mathbf{R})$.

Ясно также, что задача построения множества всех оптимальных стратегий принадлежит классу $WORK - Lcl(\mathbf{R})$, если либо \mathbf{R} – представление G_M матрицей смежности вершин, либо

$$m = \Omega(k) \quad (k \rightarrow \infty)$$

и \mathbf{R} – представление G_M либо матрицей смежности дуг, либо списками смежности (либо вершин, либо дуг). В случае, когда

$$m = o(k) \quad (k \rightarrow \infty)$$

и \mathbf{R} – представление G_M либо матрицей смежности дуг, либо списками смежности (либо вершин, либо дуг), вопрос о принадлежности задачи построения множества всех оптимальных стратегий классу $WORK - Lcl(\mathbf{R})$ – открытый. Также открытым является вопрос о принадлежности классу $WORK - Lcl(\mathbf{R})$ задачи построения множества всех не избыточных стратегий для всех представлений \mathbf{R} направленного мультиграфа G_M .

Рассмотрим теперь задачи построения стратегий идентификации множества вершин Q_0 в направленном мультиграфе G_M ($M \in \mathbf{M}_{k,m,n}$), предназначенных как для идентификации начальной вершины, так и для завершения всех прогулок с началом, принадлежащем множеству Q_0 , в одной и той же вершине.

Очевидно, что если

$$|Q_0| = 2,$$

то задача построения всех оптимальных стратегий принадлежит классу $Lcl(\mathbf{R})$ для всех представлений \mathbf{R} направленного мультиграфа G_M (однако, открытым является вопрос о принадлежности классу $WORK - Lcl(\mathbf{R})$ задачи построения множества всех не избыточных стратегий для всех представлений \mathbf{R} направленного мультиграфа G_M).

Из нижних оценок функций Шеннона для длин минимальных диагностических и синхронизирующих слов для слабоинициального автомата (M, Q_0) ($M \in \mathbf{M}_{k,2,n}$) вытекает, что задачи построения как оптимальных, так и не избыточных стратегий не принадлежат классу $WORK - Lcl(\mathbf{R})$ для всех представлений \mathbf{R} направленного мультиграфа G_M , при условии, что

$$3 \leq |Q_0| \leq |Q| - 1.$$

Однако для всех представлений \mathbf{R} направленного мультиграфа G_M открытым является вопрос о принадлежности классу $WORK - Lcl(\mathbf{R})$ задач

построения как оптимальных, так и не избыточных стратегий в случае, когда

$$Q_0 = Q.$$

Идентифицирующие для заданного слабоинициального автомата (M, Q_0) ($M \in \mathbf{M}_{k,m,n}$) слова определяют тип экспериментов с автоматом, который принято называть *безусловным*. Этот тип эксперимента характеризуется тем, что на исследуемый автомат полностью подается входное слово и наблюдается реакция автомата на него. После этого на анализе только реакции автомата на поданное на него слово делается заключение о соответствующем состоянии автомата и эксперимент завершается. В соответствии с этим, рассмотренные выше стратегии идентификации множества вершин Q_0 в направленном мультиграфе G_M ($M \in \mathbf{M}_{k,m,n}$) также естественно назвать *безусловными стратегиями*.

Другим типом эксперимента с автоматом является *условный* эксперимент. Он характеризуется тем, что для подаваемого на автомат слова каждый входной символ, начиная со второго, формируется в зависимости от реакции автомата на предыдущие символы. В 1973 году В.Г. Скобелев предложил методы построения *автоматов-экспериментаторов*, осуществляющих идентификацию состояний слабоинициального автомата (M, Q_0) ($M \in \mathbf{M}_{k,m,n}$).

Автомат-экспериментатор естественно назвать *адаптивной стратегией* идентификации множества вершин Q_0 в направленном мультиграфе G_M ($M \in \mathbf{M}_{k,m,n}$). Ясно, что прогулка по направленному мультиграфу G_M ($M \in \mathbf{M}_{k,m,n}$) используемого при адаптивной стратегии идентификации множества вершин Q_0 автомата-экспериментатора допускает естественную интерпретацию в терминах поведения автомата в конечном (не обязательно прямоугольном) лабиринте.

Для сложности построения адаптивных стратегий идентификации множества вершин Q_0 в направленном мультиграфе G_M ($M \in \mathbf{M}_{k,m,n}$) истинно все то, что было сказано выше о сложности построения соответствующих безусловных стратегий. Это непосредственно вытекает из того, что число состояний автомата-экспериментатора может совпадать с длиной минимального идентифицирующего слова.

В рассмотренных выше экспериментах с автоматами используется одно входное слово или, соответственно, один автомат-экспериментатор. Такие эксперименты принято называть *простыми*. По аналогии с этим соответствующие стратегии идентификации множества вершин Q_0 в направленном мультиграфе G_M ($M \in \mathbf{M}_{k,m,n}$) также естественно назвать *простыми*.

Если в процессе эксперимента с автоматом используется более, чем одноэлементное, множество входных слов (соответственно, более, чем одноэлементное, множество автоматов-экспериментаторов), каждый элемент которого предназначен для частичного решения задачи идентификации состояния автомата, а, в совокупности, эти частичные решения дают полное решение, то эксперимент принято называть *кратным*. Соответствующие стратегии идентификации множества вершин Q_0 в направленном мультиграфе G_M ($M \in \mathbf{M}_{k,m,n}$) естественно назвать *кооперативными*.

Ясно, что прогулки множества автоматов-экспериментаторов по направленному мультиграфу G_M ($M \in \mathbf{M}_{k,m,n}$), используемого для кооперативной адаптивной стратегии идентификации множества вершин Q_0 , допускают естественную интерпретацию в терминах поведения коллектива автоматов в конечном (не обязательно прямоугольном) лабиринте.

Следует отметить, что задача исследования сложности построения кооперативных стратегий (как безусловных, так и адаптивных) идентификации множества вершин Q_0 в направленном мультиграфе G_M ($M \in \mathbf{M}_{k,m,n}$) не решена до сих пор.

Замечание 4.2. Направленный мультиграф G_M , ассоциированный с автоматом M принято называть *автоматным*. Ясно, что определенные выше стратегии прогулок могут быть обобщены на случай произвольных направленных мультиграфов с раскрашенными дугами, изображения отметок вершин которых покрыты кляксами, при надлежащем определении характеристики прогулки.

4.3. Системы дискретных событий.

В течение последних пятнадцати лет интенсивно развивается направление, получившее имя *системы дискретных событий* (СДС). Основной предмет его исследования – это моделирование и синтез управления систем большой размерности таких, как гибкие производственные системы, компьютерные и коммуникационные сети, системы управления в экономике и т.д. Существенной характеристикой СДС является то, что СДС – динамическая система, эволюция которой управляется внезапными наступлениями тех или иных событий в нерегулярные моменты времени.

Среди различных, слабо связанных друг с другом, математических моделей СДС, большое распространение получила модель автоматного типа, которую предложили в 1987 году *P.J. Ramage* и *W.M. Wonham* (см., напр. [17]). Эта модель применяется, в основном, для построения *адаптивного управления*, т.е. для реализации систем типа *closed-loop*, обеспечивающих принадлежность *траекторий событий* заданному языку. Формально, модель автоматного типа для СДС определяется как акцептор

$$M = (Q, \Sigma, \delta, q_0, Q_m),$$

где Q и Q_m ($Q_m \subseteq Q$) – множество, соответственно, *состояний* и *маркированных состояний*, q_0 ($q_0 \in Q$) – *начальное состояние*, Σ – *алфавит отметок событий*, а $\delta : Q \times \Sigma \rightarrow Q$ – (возможно, частичная) *функция переходов*. Акцептор M характеризуется следующими двумя подмножествами множества Σ^* :

1) *замкнутым поведением*, т.е. языком

$$\mathbf{L}(M) = \{s \in \Sigma^* \mid \tilde{\delta}(q_0, s) \text{ определено}\},$$

содержащим все возможные последовательности событий;

2) *языком завершенных заданий*

$$\mathbf{L}_m(M) = \{s \in \Sigma^* \mid \tilde{\delta}(q_0, s) \in Q_m\}.$$

Предполагается, что входной алфавит Σ акцептора M разбит на два непустых множества Σ_c и Σ_{uc} , соответственно, *управляемых* и *неуправляемых* событий. Первые, при их появлении, могут быть, как реализованы, так и предотвращены (с помощью соответствующего внешнего агента). Вторые, при их появлении, реализуются обязательно.

Механизм управления предотвращением событий основан на построении супервизора

$$\mathbf{S} = (A, \varphi),$$

где:

1) $A = (U, \Sigma, \psi, u_0, U_{fin})$ – акцептор, представляющий собой настроенный автомат (т.е. ψ – отображение, определенное на множестве $U \times \Sigma$);

2) $\varphi : U \times \Sigma \rightarrow \{0,1\}$ – такое отображение, что

$$\varphi(u, \sigma) = 1,$$

если $\sigma \in \Sigma_{uc}$ и

$$\varphi(u, \sigma) \in \{0,1\},$$

если $\sigma \in \Sigma_c$, т.е. значение 1 интерпретируется как «реализовать событие», а значение 0 – как «блокировать событие». Таким образом, поведение замкнутой системы (т.е. множество последовательностей событий, реализуемых, пока СДС находится под управлением супервизора \mathbf{S}) представлено некоторой системой автоматного типа $\langle M \mid \mathbf{S} \rangle$ (см. рис. 4.2).

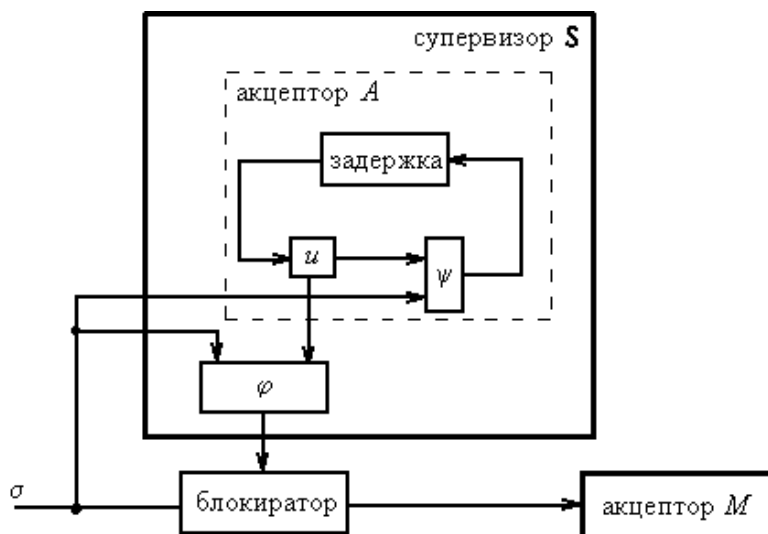


Рис. 4.2. Замкнутая система $(M|S)$.

Несмотря на значительное число работ, связанных с исследованием моделей автоматного типа для СДС, общим для них является *разработка с чистого листа*, что проявляется в игнорировании моделей и методов классической теории автоматов, хотя бы на уровне *алгебры источников*, систематически проработанной Б.А. Трахтенбротом в 1970 году. В результате возникает целый ряд *узких мест*. К наиболее существенным из них относятся следующие:

1. В основном при моделировании СДС принимается во внимание недетерминизм, связанный с предотвращением наступления управляемых событий (ради справедливости следует отметить, что в ряде работ допускается случай, когда $\delta \subseteq Q \times \Sigma \times Q$ – тернарное отношение, хотя это не решает проблемы). Таким образом, значительная часть неопределенности, присущей реальным системам, остается вне поля зрения (отметим, что эффективным средством для представления значительной части такой неопределенности являются *пустые дуги* в источнике).

2. Почти во всех работах предполагается, что при эволюции СДС отсутствуют конкурентные процессы. Фактически это означает, что принято предположение (истинность которого представляется весьма сомнительной): существующие конкурентные процессы *спрятаны* в состоянии акцептора M . Отметим, что источники дают возможность представлять конкурентные процессы на уровне подсистемы *недетерминизмом в переходах* и наличием *пустых дуг*, а на уровне всей системы – эффективным ее представлением в виде композиции источников-компонент.

3. Сложность системы $\langle M|S \rangle$ может быть весьма высокой по сравнению со сложностью акцептора M . При этом известно, что декомпозиция

супервизора является концептуально сложной проблемой. Поэтому естественно возникает вопрос о построении децентрализованного управления для системы, являющейся композицией достаточно простых подсистем. *Алгебра источников* как раз и предоставляет эффективные средства решения этой задачи.

Таким образом, для того, чтобы обойти перечисленные выше *узкие места* и в полной мере использовать модели и методы классической теории автоматов, позволяющие эффективно работать, по крайней мере, в рамках алгебры Клини, естественно следующим образом определить модель СДС в терминах *двухполюсника* – специального типа источника.

Определение 4.1. Назовем СДС такую упорядоченную пару $\mathbf{M}_\pi = (M, \pi)$, что $M = (Q, \Sigma, \delta, q_{in}, Q_m, q_{fin})$, где Q и Q_m ($Q_m \subseteq Q$) – множество, соответственно, *состояний* и *маркированных состояний*, Σ – *алфавит отметок событий*, q_{in} и q_{fin} ($q_{in}, q_{fin} \in Q$) – соответственно, *начальное* и *финальное состояние*, а тернарное отношение *переходов*

$$\delta \subseteq Q \times (\Sigma \cup \{\Lambda\}) \times Q$$

удовлетворяет следующим двум условиям

$$(q, \Lambda, q_{fin}) \in \delta \Leftrightarrow q \in Q_m, \quad (4.1)$$

$$(\forall \sigma \in \Sigma \cup \{\Lambda\})(\forall q \in Q)(q_{fin}, \sigma, q \notin \delta), \quad (4.2)$$

а $\pi = \{\Sigma_c, \Sigma_{uc}\}$ – разбиение множества Σ на множества Σ_c и Σ_{uc} , соответственно, *управляемых* и *неуправляемых* событий.

Множество всех акцепторов $M = (Q, \Sigma, \delta, q_{in}, Q_m, q_{fin})$, имеющих фиксированные k -элементное множество состояний Q и l -элементный алфавит событий Σ обозначим через $\mathbf{M}_{k,l}$.

Замечание 4.3. Нетрудно заметить, что акцептор M из определения 4.1 представляет собой обычный *двухполюсник* с ограничениями на его структуру (определяемыми соотношениями (4.1) и (4.2)), некоторые состояния которого отмечены специальными метками (т.е. *маркированы*).

Определим отношение

$$\delta^{(n)} \subseteq Q \cup (\Sigma \cup \{\Lambda\})^n \times Q \quad (n \in \mathbf{N})$$

следующим образом

$$\begin{aligned} & (\forall q_0, q_n \in Q)(\forall \sigma_1, \dots, \sigma_n \in \Sigma \cup \{\Lambda\})((q_0, \sigma_1 \dots \sigma_n, q_n) \in \delta^{(n)}) \Leftrightarrow \\ & \Leftrightarrow (\exists q_1, \dots, q_{n-1} \in Q)(\forall i \in \{1, \dots, n\})((q_{i-1}, \sigma_i, q_i) \in \delta). \end{aligned} \quad (4.3)$$

Положим

$$\tilde{\delta} = \Delta_Q \cup \left(\bigcup_{n=1}^{\infty} \delta^{(n)} \right),$$

где

$$\Delta_Q = \{(q, \Lambda, q) \mid q \in Q\}.$$

Замечание 4.4. В формуле (4.3) фигурируют слова в алфавите Σ с, возможно, «вкрапленными» в них пустыми символами Λ . Такое соглашение существенно для определения отношений $\delta^{(n)}$ ($n \in \mathbf{N}$) и $\tilde{\delta}$. Однако в тех случаях, когда объектом исследования являются именно слова, то, в соответствии с общепринятыми правилами, эти пустые символы удаляются. Такие действия не вызывают недоразумений и, в дальнейшем на них не будет акцентироваться внимание.

Замкнутое поведение и язык завершенных заданий для СДС M_π определим равенствами

$$\mathbf{L}(M_\pi) = \{s \in \Sigma^* \mid \{(q_{in}, s, q) \in \tilde{\delta} \mid q \in Q\} \neq \emptyset\},$$

$$\mathbf{L}_m(M_\pi) = \{s \in \Sigma^* \mid (\exists q \in Q_m)((q_{in}, s, q) \in \tilde{\delta})\}.$$

Замечание 4.5. Отметим, что:

1. Условия (4.1) и (4.2) дают возможность прекращать вычисления при достижении маркированного состояния.
2. Отношение переходов вида

$$\delta \subseteq Q \times (\Sigma \cup \{\Lambda\}) \times Q$$

дает возможность эффективно реализовать следующие моменты.

Во-первых, в явном виде представить неопределенность, не связанную с предотвращением наступления управляемых событий.

Во-вторых, при эволюции СДС эффективно учитывать (если в этом возникает необходимость) наличие конкурентных процессов.

В третьих, при построении супервизора реализовать две принципиально различные возможности, а именно: осуществляется или нет переход СДС при предотвращении управляемого события.

3. Применение *стандартной процедуры детерминизации источника* дает возможность перейти от любой СДС

$$M_\pi = (M, \pi)$$

к такой СДС

$$M_\pi^{\text{dem}} = (M^{\text{dem}}, \pi),$$

где M^{dem} – настроенный автомат, что M и M^{dem} эквивалентны друг другу, как источники (число состояний настроенного автомата M^{dem} может равняться 2^k , где k – число состояний двухполюсника M , причем эта оценка – достижимая). При этом

$$\mathbf{L}_m(M_\pi) = \mathbf{L}_m(M_m^{\text{dem}}).$$

Ясно, что именно СДС типа M_π^{dem} является эффективным средством для построения супервизора

$$\mathbf{S} = (M_\pi^{\text{dem}}, \varphi)$$

для СДС M_π . Отметим также, что выполнение равенства

$$\mathbf{L}(M_\pi) = \mathbf{L}(M_m^{\text{dem}})$$

может быть обеспечено за счет специальной маркировки состояния \emptyset настроенного автомата M^{dem} . Более того, эффективно разрешима задача построения супервизора, в результате действий которого реализуется некоторое максимальное (по включению) расширение языка завершенных заданий.

Действительно, эта задача сводится к построению (обычным образом) двухполюсника, представляющего максимальный язык завершенных заданий, его детерминизации и построения соответствующего отображения φ .

Более сложная ситуация имеет место при построении супервизора, в результате действий которого реализуется оптимальный (по обычному или векторному критерию) язык завершенных заданий. Эта задача сводится к решению соответствующей задачи математического программирования (соответственно, обычной или векторной оптимизации), размерность которой может равняться $O(2^k)$ ($k \rightarrow \infty$), где k – число состояний двухполюсника M .

4. Стандартная алгебра источников, разработанная *Б.А. Трахтенбротом* в 1970 году, непосредственно применима для построения сети СДС (а, следовательно, и для решения задачи построения распределенного управления сетью СДС).

Действительно, рассмотрим СДС

$$M_\pi^{(i)} = (M_i, \pi) \quad (i = 1, 2),$$

где

$$M_i = (Q^{(i)}, \Sigma, \delta^{(i)}, q_{in}^{(i)}, Q_{in}^{(i)}, q_{fin}^{(i)}) \quad (i = 1, 2)$$

такие двухполюсники, что

$$Q^{(1)} \cap Q^{(2)} = \emptyset.$$

Отметим, что истинность последнего равенства всегда легко обеспечить за счет изменения имен состояний.

Тогда:

А. Замкнутое поведение $\mathbf{L}(M_\pi^{(1)}) \cup \mathbf{L}(M_\pi^{(2)})$ и язык завершенных заданий $\mathbf{L}_m(M_\pi^{(1)}) \cup \mathbf{L}_m(M_\pi^{(2)})$ реализуются СДС $M_\pi = (M, \pi)$, где

$$M = (Q^{(1)} \cup Q^{(2)} \cup \{q_{in}, q_{fin}\}, \Sigma, \delta, q_{in}, Q_m^{(1)} \cup Q_m^{(2)}, q_{fin}),$$

причем

$$\{q_{in}, q_{fin}\} \cap (Q^{(1)} \cup Q^{(2)}) = \emptyset$$

и

$$\delta = \delta^{(1)} \cup \delta^{(2)} \cup \{(q_{in}, \Lambda, q_{in}^{(1)}), (q_{in}, \Lambda, q_{in}^{(2)}), (q_{fin}^{(1)}, \Lambda, q_{fin}), (q_{fin}^{(2)}, \Lambda, q_{fin})\}.$$

Б. Замкнутое поведение $\mathbf{L}(M_\pi^{(1)}) \cdot \mathbf{L}(M_\pi^{(2)})$ и язык завершенных заданий $\mathbf{L}_m(M_\pi^{(1)}) \cdot \mathbf{L}_m(M_\pi^{(2)})$ реализуются СДС $M_\pi = (M, \pi)$, где

$$M = (Q^{(1)} \cup Q^{(2)}, \Sigma, \delta, q_{in}^{(1)}, Q_m^{(2)}, q_{fin}^{(2)})$$

и

$$\delta = \delta^{(1)} \cup \delta^{(2)} \cup \{(q_{fin}^{(1)}, \Lambda, q_{in}^{(2)})\}.$$

В. Замкнутое поведение $\mathbf{L}(M_\pi^{(1)}) \cap \mathbf{L}(M_\pi^{(2)})$ и язык завершенных заданий $\mathbf{L}_m(M_\pi^{(1)}) \cap \mathbf{L}_m(M_\pi^{(2)})$ реализуются СДС, построение которой осуществляется в три этапа. Первый этап состоит в построении детерминированных эквивалентов M_1^{dem} и M_2^{dem} для, соответственно, двухполюсников M_1 и M_2 с помощью обычной процедуры детерминизации источников. Второй этап состоит в построении *декартового произведения с отождествлением входа* M^{dem} для настроенных автоматов M_1^{dem} и M_2^{dem} . Третий этап состоит в преобразовании настроенного автомата M^{dem} в эквивалентный ему двухполюсник.

Г. Замкнутое поведение $(\mathbf{L}(M_\pi^{(1)}))^*$ и язык завершенных заданий $(\mathbf{L}_m(M_\pi^{(1)}))^*$ реализуются СДС $M_\pi = (M, \pi)$, где

$$M = (Q^{(1)} \cup \{q_{in}, q_{fin}\}, \Sigma, \delta, q_{in}, Q_m^{(1)}, q_{fin}),$$

причем

$$\{q_{in}, q_{fin}\} \cap Q^{(1)} = \emptyset$$

и

$$\delta = \delta^{(1)} \cup \delta^{(2)} \cup \{(q_{in}, \Lambda, q_{in}^{(1)}), (q_{fin}^{(1)}, \Lambda, q_{in}^{(1)}), (q_{fin}^{(1)}, \Lambda, q_{fin})\}.$$

Д. Замкнутое поведение $\overline{\mathbf{L}(M_\pi^{(1)})}$ и язык завершенных заданий $\overline{\mathbf{L}_m(M_\pi^{(1)})}$ реализуются СДС, построение которой осуществляется в три этапа. Первый этап состоит в построении детерминированного эквивалента M_1^{dem} для двухполюсника M_1 . Второй этап состоит в изменении настройки в M_1^{dem} в соответствии со следующим правилом: множества маркированных и финальных состояний заменяются их дополнениями. Третий этап состоит в преобразовании полученного на втором этапе настроенного автомата в эквивалентный ему двухполюсник.

Отметим, что из пяти рассмотренных выше операций только операции *пересечения* и *дополнения* могут привести к существенному усложнению СДС.

Одним из способов задания СДС

$$\mathbf{M}_\pi = (M, \pi),$$

где

$$M = (Q, \Sigma, \delta, q_{in}, Q_m, q_{fin}),$$

является ее представление направленным мультиграфом (возможно, с петлями) G_M , дуги которого раскрашены элементами множества $\Sigma \cup \{\Lambda\}$ в соответствии со специальными правилами. Положим

$$(\Sigma \cup \{\Lambda\})(q, q') = \{\sigma \in (\Sigma \cup \{\Lambda\}) \mid ((q, \sigma, q') \in \delta)\},$$

$$wght(q, q') = |\{\sigma \in \Sigma \cup \{\Lambda\} \mid (q, \sigma, q') \in \delta\}| \quad (q, q' \in Q).$$

Направленный мультиграф G_M характеризуется следующим образом:

1) множеством вершин направленного мультиграфа G_M является множество Q ;

2) множество дуг направленного мультиграфа G_M содержит $|\delta|$ элементов, определяемых следующим образом: число экземпляров дуги (q, q') ($q, q' \in Q$) равно $wght(q, q')$ (равенство $wght(q, q') = 0$ означает, что дуга (q, q') вообще отсутствует), причем эти экземпляры окрашены различными элементами множества $(\Sigma \cup \{\Lambda\})(q, q')$.

Пример 4.5. На рис. 4.3 изображен направленный мультиграф G_{M_π} , представляющий такую СДС

$$\mathbf{M}_\pi = (M, \pi),$$

что

$$M = (Q, \Sigma, \delta, q_{in}, Q_m, q_{fin}),$$

где

$$Q = \{1, \dots, 6\}, \quad Q_m = \{2, 3, 5\},$$

$$q_{in} = 1, \quad q_{fin} = 6,$$

$$\Sigma = \{0, 1, 2\}, \quad \Sigma_c = \{0, 1\}, \quad \Sigma_{uc} = \{2\},$$

$$\delta = \{(1, \Lambda, 4), (1, 0, 2), (1, 1, 3), (2, 0, 2), (2, 1, 2), (2, 2, 4), (3, 2, 1), (3, \Lambda, 6), \\ (4, 0, 5), (5, 1, 5), (5, \Lambda, 6)\}.$$

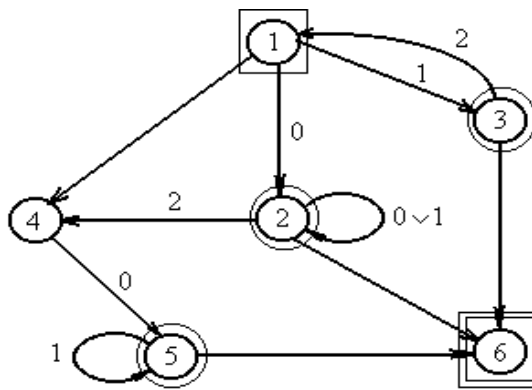


Рис. 4.3. Представление СДС $M_{\mathcal{X}}$ направленным мультиграфом.

Для упрощения изображения обычно принимается соглашение о том, что все экземпляры дуги (q, q') представляются одним экземпляром. На этом экземпляре записываются все элементы множества $(\Sigma \cup \{\Lambda\})(q, q')$, разделенные знаками дизъюнкции.

На рис. 4.3 также приняты следующие соглашения: вершины, соответствующие начальному и финальному состояниям обведены, соответственно, прямоугольником и двумя прямоугольниками, а вершины, соответствующие маркированным состояниям заключены в кружки.

Нетрудно убедиться в том, что для рассматриваемой СДС замкнутое поведение и язык завершенных заданий, записанные в виде регулярных выражений, равны

$$\mathbf{L}(M_{\pi}) = 0 \cdot (0 \vee 1)^* \cdot (\Lambda \vee 2 \cdot (\Lambda \vee 0 \cdot 1^*)) \vee (12)^* \cdot (\Lambda \vee 1 \cdot (21)^*),$$

$$\mathbf{L}_m(M_{\pi}) = 0 \cdot (0 \vee 1)^* \cdot (\Lambda \vee 20 \cdot 1^*) \vee (12)^* \cdot 1.$$

Настроенный автомат M^{dem} , эквивалентный двухполюснику M , полученный в результате стандартной процедуры детерминизации, изображен на рис.4.4, где

$$1 = \{1,4\}, \quad 2 = \{3,6\}, \quad 3 = \{2,5,6\}, \quad 4 = \{4\},$$

$$5 = \{2,6\}, \quad 6 = \{5,6\}, \quad 7 = \emptyset.$$

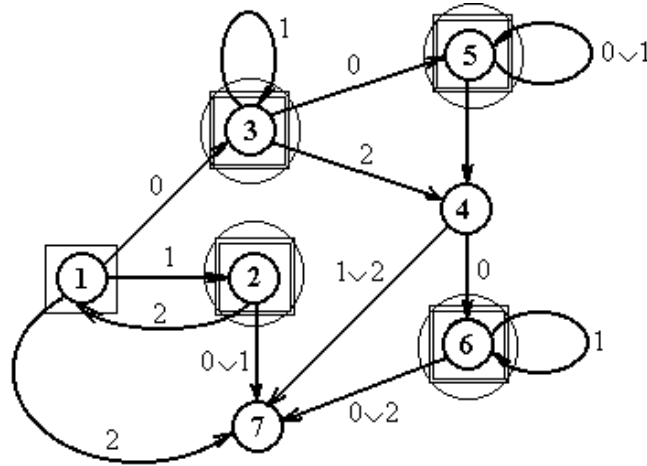


Рис. 4.4. Настроенный автомат M^{det} , эквивалентный двухполюснику M .

Этот настроенный автомат непосредственно может быть использован для построения такого супервизора

$$S = (M^{det}, \varphi),$$

что

$$\varphi \equiv 1.$$

Покажем, что задачи построения супервизора

$$S = (A, \varphi)$$

для СДС

$$M_\pi = (M, \pi) \quad (M \in \mathbf{M}_{k,l})$$

могут быть охарактеризованы в терминах задач построения адаптивных стратегий специальных прогулок по направленному мультиграфу G_M , представляющему двухполюсник

$$M = (Q, \Sigma, \delta, q_{in}, Q_m, q_{fin}),$$

изображения отметок вершин которого, за исключением вершины q_{fin} , *покрыты кляксами*.

Действительно, рассмотрим произвольный супервизор

$$\mathbf{S} = (A, \varphi)$$

для СДС

$$\mathbf{M}_\pi = (M, \pi) \quad (M \in \mathbf{M}_{k,l}).$$

Для простоты изложения предположим, что в СДС отсутствуют конкурентные процессы и неопределенность, связанная с переходами двухполюсника M по дугам, отмеченным пустым символом Λ .

Поведение системы автоматного типа $\langle M | \mathbf{S} \rangle$ ($M \in \mathbf{M}_{k,l}$) естественно интерпретируется как адаптивная стратегия *принудительной прогулки* супервизора \mathbf{S} по направленному мультиграфу G_M ($M \in \mathbf{M}_{k,l}$) в соответствии со следующими правилами:

1. Супервизору \mathbf{S} , находящемуся в вершине q_{in} последовательно, буква за буквой, предъявляется последовательность событий

$$\sigma_1 \sigma_2 \dots \sigma_n \in \Sigma^+.$$

2. Если супервизор \mathbf{S} находится в вершине q_{fin} , то все дальнейшие вычисления прекращаются, и прогулка заканчивается.

3. Пусть супервизор \mathbf{S} , после действий в соответствии с начальным отрезком последовательности событий

$$\sigma_1 \sigma_2 \dots \sigma_{i-1} \quad (i = 1, \dots, n),$$

находится в вершине q ($q \neq q_{fin}$) направленного мультиграфа G_M .

Если $\sigma_i \in \Sigma$ – такое событие, что

$$\sigma_1 \dots \sigma_{i-1} \sigma_i \notin \mathbf{L}(\mathbf{M}_\pi),$$

то супервизор \mathbf{S} прекращает все дальнейшие вычисления и прогулка заканчивается.

Пусть $\sigma_i \in \Sigma$ – такое событие, что

$$\sigma_1 \dots \sigma_{i-1} \sigma_i \in \mathbf{L}(\mathbf{M}_\pi).$$

Если

$$\varphi(u, \sigma_i) = 1,$$

где u – состояние акцептора A , то супервизор \mathbf{S} осуществляет переход по одной из дуг направленного мультиграфа G_M , исходящей из вершины q и раскрашенной событием σ_i (и, возможно, по некоторой следующей за ней последовательности дуг, раскрашенных пустым символом Λ).

Если же $\sigma_i \in \Sigma_c$ и

$$\varphi(u, \sigma) = 0,$$

то супервизор \mathbf{S} завершает действия в соответствии с начальным отрезком последовательности событий

$$\sigma_1 \sigma_2 \dots \sigma_{i-1} \sigma_i$$

и переходит к обработке следующего события.

Выше было отмечено, число состояний акцептора A супервизора

$$\mathbf{S} = (A, \varphi)$$

для СДС

$$\mathbf{M}_\pi = (M, \pi) \quad (M \in \mathbf{M}_{k,l})$$

может быть экспонентой от числа вершин направленного мультиграфа G_M . Из этого обстоятельства непосредственно вытекают следующие характеристики задач построения адаптивных стратегий принудительной прогулки по направленному мультиграфу G_M ($M \in \mathbf{M}_{k,l}$) с покрытыми кляксами вершинами.

Во-первых, задача построения адаптивной стратегии принудительной прогулки по направленному мультиграфу G_M ($M \in \mathbf{M}_{k,l}$) с покрытыми кляксами вершинами, реализующей максимальный язык завершенных заданий, не принадлежит классу $WORK - Lcl(\mathbf{R})$ для всех представлений \mathbf{R} направленного мультиграфа G_M .

Во-вторых, задача построения адаптивной стратегии принудительной прогулки по направленному мультиграфу G_M ($M \in \mathbf{M}_{k,l}$) с покрытыми кляксами вершинами, реализующей оптимальный (даже по скалярному критерию) язык завершенных заданий, также не принадлежит классу $WORK - Lcl(\mathbf{R})$ для всех представлений \mathbf{R} направленного мультиграфа G_M .

Замечание 4.6. Нетрудно заметить, что рассмотренные выше адаптивные стратегии принудительных прогулок по направленному мультиграфу G_M ($M \in \mathbf{M}_{k,l}$) с покрытыми кляксами вершинами допускают естественную интерпретацию как *стратегии игры с Природой* на направленном мультиграфе G_M

4.4. Игры на графах.

Под *игрой двух лиц* будем понимать систему

$$\mathbf{g} = (P, F, G, p_{in}, P_1^{win}, P_2^{win}), \quad (4.4)$$

где P – конечное множество позиций, $p_{in} \in P$ – исходная позиция, P_1^{win} и P_2^{win} ($P_1^{win}, P_2^{win} \neq \emptyset; P_1^{win} \cap P_2^{win} = \emptyset; p_{in} \notin P_1^{win} \cup P_2^{win}$) – множества выигрышных позиций для, соответственно, 1-го и 2-го игроков, а F и G – множества (возможно, частичных) отображений множества P в себя, называемых множествами ходов, соответственно, 1-го и 2-го игроков.

Партией в игре \mathbf{g} назовем любую такую последовательность

$$p_0^{(1)}, f_1, p_0^{(2)}, g_1, p_1^{(1)}, f_2, p_1^{(2)}, g_2, \dots, \quad (4.5)$$

что:

1) $p_0^{(1)} = p_{in}$ и

$$p_i^{(1)} \in Dom f_{i+1}, \quad p_i^{(2)} \in Dom g_{i+1}$$

для всех $i = 0, 1, \dots$;

2) как позиции

$$p_0^{(1)}, p_1^{(1)}, \dots,$$

так и позиции

$$p_0^{(2)}, p_1^{(2)}, \dots$$

попарно различны;

3) при появлении позиции

$$p \in P_1^{win} \cup P_2^{win}$$

партия прекращается.

Таким образом, все партии игры \mathbf{g} разбиваются на следующие четыре множества

$$\Pi_1^{win} = \{p_0^{(1)}, f_1, \dots, p_{k-1}^{(1)}, f_k, p_{k-1}^{(2)} \mid (k \in \mathbf{N}) \mid p_{k-1}^{(2)} \in P_1^{win}\}, \quad (4.6)$$

$$\Pi_2^{win} = \{p_0^{(1)}, f_1, \dots, p_{k-1}^{(2)}, g_k, p_k^{(1)} \mid (k \in \mathbf{N}) \mid p_k^{(1)} \in P_2^{win}\}, \quad (4.7)$$

$$\tilde{\Pi}_1 = \{p_0^{(1)}, f_1, \dots, p_{k-1}^{(2)} \mid (k \in \mathbf{N}) \mid (p_{k-1}^{(2)} \notin P_1^{win}) \& (\forall g \in G)(p_{k-1}^{(2)} \notin Dom g)\}, \quad (4.8)$$

$$\tilde{\Pi}_2 = \{p_0^{(1)}, f_1, \dots, p_k^{(1)} \mid (k \in \mathbf{N}) \mid (p_k^{(1)} \notin P_2^{win}) \& (\forall f \in F)(p_k^{(1)} \notin Dom f)\}. \quad (4.9)$$

В дальнейшем будем рассматривать только такие игры \mathbf{g} , что

$$\tilde{\Pi}_1 = \tilde{\Pi}_2 = \emptyset.$$

Это предположение не ограничивает общность рассуждений и упрощает изложение.

Стратегией 1-го игрока в игре \mathbf{g} назовем такой алгоритм A_1 , что для любой партии (4.5)

$$f_{i+1} = A_1(p_i^{(1)})$$

для всех $i = 0, 1, \dots$. Аналогичным образом, стратегией 2-го игрока в игре \mathbf{g} назовем такой алгоритм A_2 , что для любой партии (4.5)

$$g_{i+1} = A_2(p_i^{(2)})$$

для всех $i = 0, 1, \dots$.

Розыгрыш любой партии в игре \mathbf{g} двух лиц допускает следующую естественную интерпретацию в терминах вычисления словарных функций специальным вариантом двумерной 2-головочной МТ M с двумя управляющими устройствами (см. рис. 4.5).

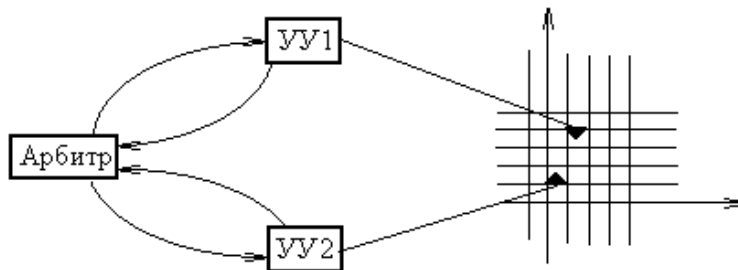


Рис. 4.5. Представление партии игры \mathbf{g} двумерной 2-головочной МТ M с двумя управляющими устройствами.

Управляющие устройства $УУ1$ и $УУ2$ (и управляемые ими головки) представляют, соответственно 1-го и 2-го игроков. При этом, $УУ1$ реализует некоторую стратегию A_1 1-го игрока, а $УУ2$ – некоторую стратегию A_2 2-го игрока. Начальная конфигурация МТ M содержит представление исходной позиции $p_{in} = p_0^{(1)}$. В начальный момент времени *арбитр* запус-

кает управляющее устройство УУ1, которое выполняет вычисления в соответствии с ходом

$$f_1 = A_1(p_{in}).$$

Закончив вычисления, управляющее устройство УУ1 сообщает об этом арбитру и останавливается в конфигурации, содержащей представление позиции $p_0^{(2)}$. Если партия не закончена, то арбитр запускает управляющее устройство УУ2, которое выполняет вычисления в соответствии с ходом

$$g_1 = A_2(p_0^{(2)}).$$

Закончив вычисления, управляющее устройство УУ2 сообщает об этом арбитру и останавливается в конфигурации, содержащей представление позиции $p_1^{(1)}$. Если партия не закончена, то арбитр запускает управляющее устройство УУ1 и т.д.

Замечание 4.7. Размерность 2 для ленты МТ выбрана только с целью наглядности, а именно: чтобы подчеркнуть, что управляющие устройства могут использовать для своих вычислений *внешнюю память* в том объеме, который им понадобится, совершенно не мешая при этом друг другу. Например, без ограничения общности можно считать, что все позиции в разыгрываемой партии игры **g** представлены в 1-м квадранте, а УУ1 и УУ2 для своих вычислений используют клетки, составляющие, соответственно, 2-й и 4-й квадранты.

Очевидно, также, что оба управляющих устройства и арбитра можно объединить в одно управляющее устройство. Однако, при этом, будет полностью потеряна наглядность интерпретации и «в тень» уйдут ряд существенных моментов, связанных с построениями стратегий игроков.

Стратегию A_1 1-го игрока в игре **g** назовем *выигрышной*, если при ее применении 1-м игроком при любых допустимых стратегиях A_2 2-го игрока разыгрываются только партии, принадлежащие множеству Π_1^{win} . Аналогичным образом, стратегия A_2 2-го игрока – *выигрышная*, если при ее применении 2-м игроком при любых допустимых стратегиях A_1 1-го игрока разыгрываются только партии, принадлежащие множеству Π_2^{win} .

Из приведенной выше интерпретации партии игры **g** в терминах МТ M вытекает следующее.

Задача построения выигрышной стратегии для 1-го игрока (соответственно, для 2-го игрока) эквивалентна задаче построения такого управляющего устройства УУ1 (соответственно, управляющего устройства УУ2), что для любого допустимого управляющего устройства УУ2 (соответственно, управляющего устройства УУ1) МТ M выполняет вычисления, являющиеся представлением партии, принадлежащей множеству Π_1^{win} (соответственно, принадлежащей множеству Π_2^{win}).

Ясно, что каждое из свойств «быть выигрышной стратегией 1-го игрока» и «быть выигрышной стратегией 2-го игрока» эквивалентно существованию положительного решения последовательности задач БФК. Отсюда, в частности вытекает, что часто построение выигрышной стратегии для игрока (как 1-го, так и 2-го) эквивалентно построению управляющего устройства МТ M (соответственно, $УУ1$ или $УУ2$), имеющего экспоненциальные размеры по сравнению с «размерностью» игры g (фактически это означает, что дерево игры не может быть «свернуто» в автомат с числом состояний порядка «размерности» игры g).

Замечание 4.8. Естественным обобщением рассмотренной выше игры 2-х лиц является игра h лиц ($h \in \mathbf{N}, h \geq 3$), в том числе и *коалиционная игра*. Розыгрыш партии в такой игре естественно интерпретируется в терминах вычисления словарных функций специальным вариантом двумерной h -головочной МТ M с h управляющими устройствами (причем устройства, соответствующие игрокам, входящим в *коалицию*, имеют возможность «обмениваться информацией»).

Все сказанное выше истинно для достаточно широкого класса игр двух лиц на графе (или направленном графе, мультиграфе и т.д.) G . В таких играх начальная конфигурация содержит представление R графа G (как правило, либо списками, либо матрицами смежности (либо вершин, либо ребер)), а целью игроков, как правило, является построение на графе G взаимно исключающих объектов.

В дальнейшем, для определенности, будем рассматривать задачу построения выигрышной стратегии для 1-го игрока в игре двух лиц на графе. Это предположение не ограничивает общности рассуждений, а только упрощает изложение.

Замечание 4.9. В терминах игры двух лиц на графе естественно формулируется широкий класс прикладных задач, в которых исследуемый объект моделируется графом (или направленным графом, мультиграфом и т.д.). В таких задачах, как правило, цель 1-го игрока состоит в том, чтобы обеспечить для исследуемого объекта выполнение требуемого свойства, сформулированного в терминах теории графов, а 2-й игрок представляет дестабилизирующие воздействия (т.е. возникновение различного типа неисправностей, конкурентных процессов и т.д.), препятствующих достижению поставленной цели.

Таким образом, задача построения выигрышной стратегии для 1-го игрока состоит в том, чтобы обеспечить для исследуемого объекта требуемое свойство в условиях дестабилизирующих воздействий (возможно, внешней среды).

Задача построения выигрышной стратегии для 1-го игрока в игре g двух лиц на графе принадлежит классу $P - SPACE - COMPLETE$, если утверждение о том, что 1-й игрок имеет *форсированный выигрыш* за k ходов (k – нечетное число) формализуется в виде следующей *не укорачиваемой формулы* (см., напр., [3]):

существует такой ход 1-го игрока из позиции $p_0 = p_{in}$ в позицию p_1 ,
 что
 для любого хода 2-го игрока из позиции p_1 в позицию p_2 ,
 существует такой ход 1-го игрока из позиции p_2 в позицию p_3 ,
 что
 для любого хода 2-го игрока из позиции p_3 в позицию p_4 ,

 существует такой ход 1-го игрока из позиции p_{k-3} в позицию p_{k-2} ,
 что
 для любого хода 2-го игрока из позиции p_{k-2} в позицию p_{k-1}
 существует такой ход 1-го игрока из позиции p_{k-1} в такую позицию p_k ,
 что

$$p_k \in P_1^{win}.$$

В этой формуле содержится последовательность чередующихся кванторов \exists и \forall , что, как известно, и приводит к классической $P - SPACE - COMPLETE$ задаче БФК.

Замечание 4.10. Существенное отличие стратегии игрока в игре 2-х лиц на графе G от стратегии прогулки по графу G (например, от тех стратегий прогулок, которые были рассмотрены в пп. 4.2 и 4.3) состоит в следующем.

Во-первых, стратегия прогулки по графу использует только *обход графа G* , в то время как стратегия игрока в игре 2-х лиц на графе, при необходимости, может использовать также тот или иной *обход представления \mathbf{R} графа G* .

Во-вторых, стратегия прогулки по графу, при ее реализации, не использует внешнюю память для «вычислений», в то время как стратегия игрока в игре 2-х лиц на графе может, при необходимости, использовать для «вычислений» потенциально неограниченную память.

Ясно, что для того чтобы убедиться, что задача построения выигрышной стратегии для 1-го игрока в некоторой игре \mathbf{g} двух лиц на графе $G = (V, E) \in \mathbf{G}(n, m)$ не принадлежит классу $WORK - Lcl(\mathbf{R})$ для всех представлений \mathbf{R} графа G , достаточно показать, что эта задача принадлежит классу $P - SPACE - COMPLETE$. Рассмотрим несколько примеров таких задач.

Замечание 4.11. 1. Как правило, постановка задачи построения выигрышной стратегии для 1-го игрока в игре двух лиц на графе представлена на неформальном уровне (т.е. в дескриптивной форме на обычном разговорном языке, возможно, с использованием некоторой терминологии теории графов).

Для того чтобы сформулировать такую задачу в терминах математической модели игры двух лиц (например, в виде системы (4.4)), к которой применим анализ на принадлежность классу $P - SPACE - COMPLETE$, достаточно:

- 1) определить *множество допустимых позиций*;
- 2) определить *критерий* выявления игрока, имеющего *ход* в данной позиции;
- 3) определить *способ изменения позиции* в результате хода;
- 4) указать *заключительные позиции* и победителя;
- 5) указать *исходную позицию*.

2. В приведенных ниже примерах рассматривается (обычный неориентированный) граф, а не направленный граф, или мультиграф и т.д. Это предположение никак не влияет на общность рассуждений и выбрано лишь для удобства изложения, так как терминология для различных типов графов – различная.

Пример 4.6. Покажем, что простейшие задачи построения пути между двумя фиксированными вершинами графа $G = (V, E) \in \mathbf{G}(n, m)$ при наличии дестабилизирующих воздействий внешней среды, связанных с запретом выбора либо вершин, либо ребер, не принадлежат классу $WORK - LC(\mathbf{R})$ для всех представлений \mathbf{R} графа G .

1. Одна из наиболее известных $P - SPACE - COMPLETE$ задач построения выигрышной стратегии для 1-го игрока в игре двух лиц на графе – **ОБОБЩЕННЫЙ ГЕКС** (см., напр., [3]). Постановка этой задачи, как правило, имеет следующий вид

Условие: Выделены две вершины v' и v'' графа $G = (V, E)$. Два игрока ходят по очереди. Каждый игрок в процессе своего хода выбирает одну из неокрашенных вершин графа G и окрашивает ее (1-й игрок окрашивает выбранную вершину в белый цвет, а 2-й игрок – в черный цвет). Цель 1-го игрока состоит в том, чтобы построить $(v' - v'')$ -путь, все вершины которого окрашены в белый цвет. Если такой путь построен, то выиграл 1-й игрок, в противном случае выиграл 2-й игрок.

Вопрос: Существует ли выигрышная стратегия для 1-го игрока?

Построим для рассматриваемой задачи математическую модель игры двух лиц на графе.

Назовем *позицией* упорядоченную тройку (V_1, V_2, V_3) , где V_1, V_2, V_3 – такие попарно непересекающиеся подмножества множества V , что

$$V_1 \cup V_2 \cup V_3 = V.$$

Исходная позиция имеет вид $(\emptyset, \emptyset, V)$.

Заключительными позициями являются все такие позиции (V_1, V_2, V_3) , что подграф графа G , индуцированный множеством вершин V_1 , содержит $(v' - v'')$ -путь, а также все такие позиции (V_1, V_2, V_3) , что $V_3 = \emptyset$.

В *заключительной* позиции 1-й игрок выигрывает тогда и только тогда, когда подграф графа G , индуцированный множеством вершин V_1 , содержит $(v' - v'')$ -путь.

Если (V_1, V_2, V_3) – такая не заключительная позиция, что $|V_1| + |V_2|$ – четное число, то ход 1-го игрока. Если же (V_1, V_2, V_3) – такая не заключительная позиция, что $|V_1| + |V_2|$ – нечетное число, то ход 2-го игрока.

Пусть (V_1, V_2, V_3) – текущая позиция, не являющаяся заключительной.

Ход 1-го игрока состоит в выборе вершины $v \in V_3$ и замене позиции (V_1, V_2, V_3) позицией

$$(V_1 \cup \{v\}, V_2, V_3 \setminus \{v\}),$$

а ход 2-го игрока – в выборе вершины $v \in V_3$ и замене позиции (V_1, V_2, V_3) позицией

$$(V_1, V_2 \cup \{v\}, V_3 \setminus \{v\}).$$

Для построенной математической модели утверждение о том, что 1-й игрок имеет форсированный выигрыш за k ходов (k – нечетное число) формализуется в виде не укорачиваемой формулы (так как «сворачивание» дерева игры осуществимо только на основании отношения равенства отметок его вершин), содержащей последовательность чередующихся кванторов \exists и \forall .

Следовательно, рассматриваемая задача построения выигрышной стратегии для 1-го игрока в игре двух лиц на графе $G = (V, E) \in \mathbf{G}(n, m)$ принадлежит классу $P - SPACE - COMPLETE$.

2. Реберный аналог игры ОБОБЩЕННЫЙ ГЕКС имеет следующий вид

Условие: Выделены две вершины v' и v'' графа $G = (V, E)$. Два игрока ходят по очереди. Каждый игрок в процессе своего хода выбирает одно из неокрашенных ребер графа G и окрашивает ее (1-й игрок окрашивает выбранное ребро в белый цвет, а 2-й игрок – в черный цвет). Цель 1-го игрока состоит в том, чтобы построить $(v' - v'')$ -путь, все ребра которого окрашены в белый цвет. Если такой путь построен, то выиграл 1-й игрок, в противном случае выиграл 2-й игрок.

Вопрос: Существует ли выигрышная стратегия для 1-го игрока?

Построим для рассматриваемой задачи математическую модель игры двух лиц на графе.

Назовем *позицией* упорядоченную тройку (E_1, E_2, E_3) , где E_1, E_2, E_3 – такие попарно непересекающиеся подмножества множества E , что

$$E_1 \cup E_2 \cup E_3 = E.$$

Исходная позиция имеет вид $(\emptyset, \emptyset, E)$.

Заключительными позициями являются все такие позиции (E_1, E_2, E_3) , что подграф графа G , порождаемый множеством ребер E_1 , содержит $(v' - v'')$ -путь, а также все такие позиции (E_1, E_2, E_3) , что $E_3 = \emptyset$.

В заключительной позиции 1-й игрок выигрывает тогда и только тогда, когда подграф графа G , порождаемый множеством ребер E_1 , содержит $(v' - v'')$ -путь.

Если (E_1, E_2, E_3) – такая не заключительная позиция, что $|E_1| + |E_2|$ – четное число, то ход 1-го игрока. Если же (E_1, E_2, E_3) – такая не заключительная позиция, что $|E_1| + |E_2|$ – нечетное число, то ход 2-го игрока.

Пусть (E_1, E_2, E_3) – текущая позиция, не являющаяся заключительной.

Ход 1-го игрока состоит в выборе ребра $e \in E_3$ и замене позиции (E_1, E_2, E_3) позицией

$$(E_1 \cup \{e\}, E_2, E_3 \setminus \{e\}),$$

а ход 2-го игрока – в выборе ребра $e \in E_3$ и замене позиции (E_1, E_2, E_3) позицией

$$(E_1, E_2 \cup \{e\}, E_3 \setminus \{e\}).$$

Для построенной математической модели утверждение о том, что 1-й игрок имеет *форсированный выигрыш* за k ходов (k – нечетное число) формализуется в виде *не укорачиваемой формулы* (так как «сворачивание» дерева игры осуществимо только на основании отношения равенства отметок его вершин), содержащей последовательность чередующихся кванторов \exists и \forall .

Следовательно, рассматриваемая задача построения выигрышной стратегии для 1-го игрока в игре двух лиц на графе $G = (V, E) \in \mathbf{G}(n, m)$ принадлежит классу $P - SPACE - COMPLETE$.

Пример 4.7. Покажем, что задача построения *вершинного покрытия* графа $G = (V, E) \in \mathbf{G}(n, m)$ при наличии дестабилизирующих воздействий внешней среды, связанных с запретом выбора вершин, не принадлежат классу $WORK - Lcl(\mathbf{R})$ для всех представлений \mathbf{R} графа G . Постановка этой задачи в терминах построения выигрышной стратегии для 1-го игрока в игре двух лиц на графе G имеет следующий вид

Условие: Два игрока ходят по очереди. Каждый игрок в процессе своего хода выбирает одну из неокрашенных вершин графа G и окрашивает ее (1-й игрок окрашивает выбранную вершину в белый цвет, а 2-й игрок – в черный цвет). Цель 1-го игрока состоит в том, чтобы построить такое множество вершин V' графа G , окрашенных в белый цвет, что для каждого ребра $e \in E$ графа G , хотя бы один конец ребра e принадлежит множеству V' . Если такое множество вершин построено, то выиграл 1-й игрок, в противном случае выиграл 2-й игрок.

Вопрос: Существует ли выигрышная стратегия для 1-го игрока?

Построим для рассматриваемой задачи математическую модель игры двух лиц на графе.

Назовем *позицией* упорядоченную тройку (V_1, V_2, V_3) , где V_1, V_2, V_3 – такие попарно непересекающиеся подмножества множества V , что

$$V_1 \cup V_2 \cup V_3 = V.$$

Исходная позиция имеет вид $(\emptyset, \emptyset, V)$.

Заключительными позициями являются все такие позиции (V_1, V_2, V_3) , что множество вершин V_1 содержит такое подмножество вершин V' , что для каждого ребра $e \in E$ графа G , хотя бы один конец ребра e принадлежит множеству V' , а также все такие позиции (V_1, V_2, V_3) , что $V_3 = \emptyset$.

В заключительной позиции 1-й игрок выигрывает тогда и только тогда, когда множество вершин V_1 содержит такое подмножество вершин V' , что для каждого ребра $e \in E$ графа G , хотя бы один конец ребра e принадлежит множеству V' .

Если (V_1, V_2, V_3) – такая не заключительная позиция, что $|V_1| + |V_2|$ – четное число, то ход 1-го игрока. Если же (V_1, V_2, V_3) – такая не заключительная позиция, что $|V_1| + |V_2|$ – нечетное число, то ход 2-го игрока.

Пусть (V_1, V_2, V_3) – текущая позиция, не являющаяся заключительной.

Ход 1-го игрока состоит в выборе вершины $v \in V_3$ и замене позиции (V_1, V_2, V_3) позицией

$$(V_1 \cup \{v\}, V_2, V_3 \setminus \{v\}),$$

а ход 2-го игрока – в выборе вершины $v \in V_3$ и замене позиции (V_1, V_2, V_3) позицией

$$(V_1, V_2 \cup \{v\}, V_3 \setminus \{v\}).$$

Для построенной математической модели утверждение о том, что 1-й игрок имеет *форсированный выигрыш* за k ходов (k – нечетное число) формализуется в виде формулы, эквивалентной БФК.

Действительно, каждая вершина (V_1, V_2, V_3) дерева игры может быть отмечена таким множеством ребер E' графа G , что для каждого ребра $e \in E'$, хотя бы один конец ребра e принадлежит множеству V_1 . Это означает, что «чистка» дерева игры осуществима в терминах отношения частичного порядка на множестве отметок вершин, определяемого отношением включения множеств. Однако, не сложно построить последовательность графов, для которых дерево игры, полученное в результате «чистки» представимо в виде *не укорачиваемой формулы*, содержащей последовательность чередующихся кванторов \exists и \forall .

Следовательно, рассматриваемая задача построения выигрышной стратегии для 1-го игрока в игре двух лиц на графе $G = (V, E) \in \mathbf{G}(n, m)$ принадлежит классу $P - SPACE - COMPLETE$.

4.5. Выводы.

Полученные в настоящем разделе результаты показывают, что достаточно представительный класс задач дискретной математики и ее приложений, допускающих естественную формулировку в терминах теории графов, не принадлежит классу $WORK - Lcl(\mathbf{R})$ для всех представлений \mathbf{R} графа $G = (V, E) \in \mathbf{G}(n, m)$. При этом, существенным является следующие обстоятельства.

Во-первых, классу $WORK - Lcl(\mathbf{R})$ не принадлежит ряд модельных задач, связанных с построением (безусловной, адаптивной или кооперативной) стратегии про-

гулки по направленному мультиграфу (возможно, с петлями), связанной с идентификацией покрытых кляксами вершин. Естественная интерпретация адаптивной и кооперативной стратегий прогулки по направленному мультиграфу в терминах автоматов-экспериментаторов свидетельствует о высокой сложности решения задач, связанных с вычислением автоматом или коллективом автоматов предикатов, определенных на графах. Следует отметить, что последним задачам в настоящее время уделяется значительное внимание, как в фундаментальных, так и прикладных исследованиях.

Во-вторых, классу $WORK - Lcl(\mathbf{R})$ не принадлежит ряд модельных задач, связанных с построением супервизора, осуществляющего адаптивное управление системой дискретных событий автоматного типа. По видимости (учитывая прикладной характер этих задач), перспективной является разработка стратегий локального поиска (см., напр. [16]) для построения приближенных решений.

В третьих, классу $WORK - Lcl(\mathbf{R})$ не принадлежит ряд модельных задач, связанных с построением выигрышной стратегии в игре двух лиц на графе (не говоря уже об играх h лиц ($h \in \mathbf{N}, h \geq 3$) и, тем более о коалиционных играх). Значительное число не принадлежащих классу $WORK - Lcl(\mathbf{R})$ модельных задач, связанных с построением выигрышной стратегии в игре двух лиц на графе, генерируется соответствующей переформулировкой NP -полных задач теории графов (достаточно длинный список которых приведен в [3]), по аналогии с тем, как это было сделано в примере 4.7.

В четвертых, интерпретация розыгрыша партии игры в терминах вычислений многоголовочной МТ с несколькими независимыми друг от друга управляющими устройствами стимулирует интерес к исследованию этой базовой для параллельных вычислений математической модели.

ЗАКЛЮЧЕНИЕ

В монографии исследована разрешимость задач теории графов в классах алгоритмов с линейной емкостной сложностью и с линейной сложностью рабочей памяти. Актуальность этой проблемы определяется как недостаточным вниманием к алгоритмическому подходу в теории графов, так и многочисленными применениями графов в качестве математических моделей и/или структур данных в процессах разработки и анализа алгоритмов решения прикладных задач.

Основные результаты состоят в следующем:

1. В терминах емкостной сложности представления графа определены и исследованы два достаточно представительных класса задач теории графов, а именно задачи, разрешимые в классе алгоритмов с линейной емкостной сложностью и задачи, разрешимые в классе алгоритмов с линейной сложностью рабочей памяти.
2. Разработанный подход дал возможность заложить основы емкостного аналога достаточно проработанной теории NP -полноты (основное внимание в которой сосредоточено на исследовании временной сложности алгоритмов), значимость которого существенно возрастает в связи с появлением новых парадигм вычислений, таких, как ДНК-вычисления и квантовые вычисления.
3. Проведенный анализ достаточно мощных алгебраических систем графов и направленных графов дал возможность установить ряд оценок вида «в среднем ...» и «для почти всех ...» для сложности выполнения операций и проверки отношений при различных представлениях графов и направленных графов. Из этих оценок вытекает, что для рассматриваемых алгебраических систем представления матрицами и соответствующими списками смежности, с точки зрения сложности выполнения операций и проверки отношений, эквивалентны как в среднем, так и почти во всех случаях. В тоже время, выделены классы графов и направленных графов, для которых представление списками смежности предпочтительнее представления соответствующими матрицами.
4. Систематически исследованы задачи построения основных объектов на графе (а именно: всех основных типов путей, циклов и остовных деревьев). Установлены нетривиальные экспоненциальные нижние оценки функций Шеннона для количеств указанных объектов. Из этих оценок, в частности, вытекает, что задача построения одного объекта разрешима в классе алгоритмов с линейной емкостной сложностью, в то время как задача построения всех таких объектов на графе, как правило, разрешима только в классе алгоритмов с линейной сложностью рабочей памяти.
5. Для многомерного ориентированного куба исследована задача построения rebbling-процесса, т.е. специального обхода, определяемого в рамках аксиоматики PERT, ациклического орграфа с использованием камешков для покрытия проходимых вершин. Предложенная схема построения rebbling-процесса основана на рекурсии и дает возможность осуществлять rebbling-процесс с достаточно низкой относительной сложностью, измеряемой как число используемых камешков, приходящихся на одну вершину куба.

6. Показана неразрешимость в классах рассматриваемых алгоритмов ряда модельных задач дискретной математики (идентификация состояний конечного автомата, построение супервизора для автоматной модели системы дискретных событий).
7. Показано, что задача идентификация состояний конечного автомата сводится к задаче построения стратегий прогулок (в том числе совершаемых автоматом или коллективом автоматов) по направленному мультиграфу, отметки вершин которого покрыты кляксами.
8. Показано, что задача построение супервизора для автоматной модели системы дискретных событий сводится к задаче построения стратегий принудительных прогулок, совершаемых автоматом по направленному мультиграфу
9. Показана неразрешимость в классах рассматриваемых алгоритмов ряда модельных задач построения выигрышной стратегии в игре 2-х лиц на графе.

Полученные результаты являются основой для дальнейших исследований. Эти исследования могут проводиться в следующих направлениях.

Во-первых, это исследование иерархии классов задач на графах, разрешимых в классах алгоритмов с полиномиальной или экспоненциальной емкостной сложностью, а также с соответствующей сложностью рабочей памяти.

Во-вторых, это исследование сложности, точности и эффективности стратегий локального поиска для приближенного построения оптимальных объектов на графе.

В третьих, это исследование сложности rebbling-процессов для специальных классов ациклических орграфов.

В четвертых, это детальное исследование задач построения стратегий прогулок (в том числе совершаемых автоматом или коллективом автоматов) различных типов по графу.

В пятых, это исследование свойств машины Тьюринга с арбитром, головки которой управляются независимыми управляющими устройствами. Такая машина Тьюринга является одной из наиболее общих моделей параллельных вычислений и дает возможность адекватно исследовать сложность решения широкого класса задач, связанных с построением распределенного адаптивного управления системой с сетевой структурой. В частности, задач сводящихся к построению выигрышных стратегий в играх на графах.

ЛИТЕРАТУРА

1. Ахо А., Хопкрофт Дж., Ульман Дж. Построение и анализ вычислительных алгоритмов. – М.: Мир, 1979. – 536с.
2. Берж К. Теория графов и ее применения. – М.: ИЛ, 1962. – 319с.
3. Гэри М., Джонсон Д. Вычислительные машины и труднорешаемые задачи. – М.: Мир, 1982. – 416с.
4. Кормен Т., Лейзерсон Ч., Ривест Р. Алгоритмы: построение и анализ. – М.: МЦНМО, 2000. – 960с.
5. Кристофидес Н. Теория графов. Алгоритмический подход. – М.: Мир, 1978. – 432с.
6. Кудрявцев В.Б., Ушчумлич Ш., Килибарда Г. О поведении автоматов в лабиринтах // Дискретная математика. – 1992. – 4. – № 3. – С. 3-28.
7. Майника Э. Алгоритмы оптимизации на графах и сетях. – М.: Мир, 1981. – 323с.
8. Оре О. Теория графов. – М.: Наука, 1968. – 352с.
9. Рейнгольд Э., Нивергельт Ю., Део Н. Комбинаторные алгоритмы. Теория и практика. – М.: Мир, 1980. – 476с.
10. Скобелев В.Г. Анализ дискретных систем. – Донецк: ИПММ НАНУ, 2002. – 172с.
11. Трахтенброт Б.А. Сложность алгоритмов и вычислений. – Новосибирск: НГУ, 1967. – 257с.
12. Харари Ф. Теория графов. – М.: Мир, 1973. – 300с.
13. Харари Ф., Палмер Э. Перечисление графов. – М.: Мир, 1977. – 324с.
14. Bollobás B. Modern graph theory. – Springer-Verlag New York, Inc. 1998. – 394p.
14. Paun G., Rosenberg G., Salomaa A. DNA computing. Springer-Verlag, 1998. – 402p.
15. Stützle T.G. Local search algorithms for combinatorial problems – analysis, algorithms, and new applications. – St. Augustin: Infix, 1999. – 203p.
16. Wonham W.M., Ramage P.J. On the supernal controllable sublanguage of a given language // SIAM Journal of Control and Optimization. – 1987. – 25. – pp. 637-659.

Владимир Геннадиевич Скобелев

ЛОКАЛЬНЫЕ АЛГОРИТМЫ НА ГРАФАХ

ISBN 966-02-2987-9

Подписано к печати 9.07.2003. Формат 60x84 1/16.
Усл. Печ. л. 14. Печать лазерная. Заказ 296. Тираж 150 экз.

**Отпечатано в типографии ООО «Норд Компьютер»
на цифровом лазерном издательском комплексе Rank Xerox Docu Tech 135
Адрес: г. Донецк, б. Пушкина, 23. Тел.: (062)-342-14-18.**