ОЛІМПІАДА З ІНФОРМАТИКИ У МІСТІ КИЄВІ У 2012–2013 НАВЧАЛЬНОМУ РОЦІ

Продовження, початок у №8 за 2013 рік

Знов'як Юрій Володимирович,
інженер з програмного забезпечення Google New York Center,
Мисак Данило Петрович,
керівник гуртка СШ №52 м. Києва,
Рибак Олександр Владиславович,
аспірант Інституту математики НАН України,
Рудик Олександр Борисович,
кандидат фізико-математичних наук, доцент кафедри методики природничо-математичної освіти і технологій Інституту післядипломної педагогічної освіти Київського університету імені Бориса Грінченка.

6. Портали

Максимальна оцінка: 100 балів
Обмеження на час: 1,2 сек.
Обмеження на пам'ять: 250 МБ
Вхідний файл: portals.in
Вихідний файл: portals.out
Програма: portals.*

Деякі планети галактики, куди нещодавно переїхав Петрик П'яточкін, сполучено порталами. Якщо планети *А* і *Б* сполучено, це означає, що на кожній із планет стоїть спеціальний пристрій — портал — для телепортації між ними. Істота, що входить у портал на планеті *А*, миттєво опиняється на планеті *Б* і навпаки.

Один і той самий портал не можна використовувати для телепортації на різні планети. Якщо між парою планет ще немає сполучення, їх можна сполучити, але лише побудувавши на кожній із них по новому порталу. Будівництво ж порталів вимагає чималих витрат і може коштувати по-різному на різних планетах.

Будемо казати, що між двома планетами існує шлях телепортації, якщо з однієї планети можна потрапити на іншу, телепортувавшись один або кілька разів (використовуючи проміжні планети). На жаль, поки що не між кожними двома планетами існує шлях телепортації.

Завдання. Маючи схему наявного сполучення планет і знаючи вартість будівництва порталу на кожній планеті, визначте, яку найменшу суму грошей потрібно витратити, щоб забезпечити існування шляху телепортації між кожною парою планет галактики.

Вхідні дані. У першому рядку вхідного файлу вказано два натуральних числа:

n — кількість планет галактики, $n \leqslant 1000$;

m — кількість пар планет, безпосередньо сполучених між собою порталами.

У другому рядку перераховані n натуральних чисел $p_1, p_2, \ldots, p_n$ — вартості будівництва нового порталу відповідно на першій, другій, ..., n-й планеті. Жодна з вартостей не перевищує 10^6.

У кожному з наступних m рядків записано по два натуральних числа a_i та b_i, $1 \leqslant a_i < b_i \leqslant n$, $1 \leqslant i \leqslant m$, які задають пару сполучених між собою планет. Кожну пару записано у вхідному файлі лише один раз.

Вихідні дані. Вихідний файл повинен містити єдине натуральне число — *найменшу* суму грошей, яку потрібно витратити на будівництво на деяких планетах порталів, щоб між кожними двома планетами, між якими не було шляху телепортації, такий шлях було створено.

Приклад

portals.in	portals.out
4 2 7 4 7 3 1 3 2 4	10

Пояснення. Можна сполучити четверту планету або з першою, або з третьою, задовольнивши умову задачі і витративши 3+7=10 грошових одиниць. Легко зрозуміти, що меншою кількістю витрачених грошей обійтися не вдасться.

ІДЕЇ РОЗВ'ЯЗАННЯ ЗАВДАНЬ III ЕТАПУ

1. Поліклініка (автор — Данило Мисак)

Якщо $a_1 \geqslant t_1$, то Петрик може прийти у першу ж хвилину, коли прийматиме лікар, і відразу потрапити на прийом. У цьому випадку ми просто виводимо t_1. Тому далі вважатимемо, що $a_1 < t_1$.

Зрозуміло, що Петрику немає сенсу приходити до моменту часу a_1, бо інакше він потрапить на прийом тоді ж, коли потрапив би, якби прийшов у момент a_1 (а саме в момент часу t_1), але прочекає при цьому довше. Якщо хлопець прийде у момент t_2 або пізніше, він узагалі не потрапить на прийом. Таким чином, маємо $t_2 - a_1$ варіантів (a_1, a_1+1, ..., t_2-1), коли може прийти хлопець. Перше, що спадає на думку, — перебрати всі такі варіанти та для кожного моменту t визначити, чи потрапить Петрик на прийом, якщо прийде у цей момент, а якщо потрапить, то скільки йому доведеться чекати. Потім вивести оптимальний варіант.

Позначимо через $s(t)$ момент часу, коли хлопець потрапить на прийом за умови приходу у поліклініку у момент t. Почнемо перебір із варіанту $t=a_1$ у порядку зростання t до величини $t=t_2$. При $t=a_1$ маємо: $s(t)=t_1$. При переході від моменту приходу t до моменту $t+1$ маємо:

- Якщо жоден пацієнт не прийшов у момент t, то Петрик, прийшовши на хвилину пізніше у момент $t+1$, ні одного нового пацієнта перед собою не пропустить:
  - при $s(t)=t$ маємо: $s(t+1)=t+1$ — вхід вільний;
  - при $s(t)>t$ маємо: $s(t+1)=s(t)$ — час потрапляння на прийом залишиться незмінним.

  Таким чином, $s(t+1)=\max(s(t), t+1)$.
- Якщо якийсь пацієнт прийшов у момент t, тобто $t=a_k$ для деякого k, то, прийшовши на хвилину пізніше, Петрик пропустить його вперед. За умовою цей пацієнт перебував на прийомі b_k хвилин. Маємо: $s(t+1)=s(t)+b_k$.

При справдженні нерівності $s(t+1)\geqslant t_2$ перебір можна припинити (тепер хлопець уже не зможе потрапити на прийом). Насправді перебір навіть треба перервати, щоб не спричинити переповнення типу даних.

Після закінчення перебору найменша з різниць $s(t)-t$ для всіх розглянутих величин t є найменшим часом очікування, а найменше відповідне значення t — шуканим моментом часу.

Зауважимо: замість того, щоб шукати на кожному кроці пацієнта, який прийшов у момент часу t, достатньо одночасно зі збільшенням величини t рухатися масивом моментів приходу пацієнтів і зберігати у пам'яті величину k, для якої $a_k\leqslant t<a_{k+1}$, та збільшувати її час від часу на 1. Час роботи алгоритму складе в такому випадку $O(t_2-a_1)$.

Щоб досягнути ефективності $O(n)$, можна помітити, що проміжки між двома сусідніми значеннями a_k і a_{k+1} можна «перескакувати». Інакше кажучи, ми будемо розглядати не моменти $t=a_1, a_1+1, a_1+2, \ldots$, а моменти $t=a_1, a_2, a_3, \ldots$, використовуючи рівність:

$$s(a_{k+1})=s(a_k)+b_k.$$

Розгляд величин t потрібно припинити, якщо справджується одне з тверджень:

а) $s(a_k)+b_k\geqslant t_2$;

б) $s(a_k)+b_k\leqslant a_{k+1}$ при $k<n$;

в) $s(a_k)+b_k<t_2$ при $k=n$.

Якщо справджується одна з умов б) або в), відповіддю буде число $s(a_k)+b_k$. Час очікування хлопця у цьому випадку дорівнюватиме нулю.

2. Фарбування (автор — Олександр Рудик)

Рівень 1 (10 балів). При $m=1$ кількість розфарбувань дорівнює 1.

Рівень 2 (84 бали). Алгоритм розв'язання, що не опирається на теорему Редфілда-Пойя, такий:

1. Зчитавши вхідні дані, встановити кількість граней і відношення суміжності.

2. За допомогою оптимізованого перебору перестановок номерів граней встановити групу симетрій — взаємно однозначних відображень множини граней многогранника у себе, що зберігають відношення суміжності (наявності спільного ребра).

Примітка. *Кроки 1 і 2 реалізовано у розв'язанні задачі 16.2 відбірково-тренувальних зборів (Многогранник–2, III етап 2005 року), авторське розв'язання є в архіві тестових файлів.*

3. За допомогою структури даних, яку опрацьовують побітово, створити перелік цілих чисел від 0 до m^n-1, де n — кількість граней.

Примітка. *Для мови Pascal таку струкутуру можна подати масивом множин. Кожному такому цілому числу взаємно однозначно відповідає його запис довжини n у системі числення з основою m з дозволом писати нулі на початку запису. А запис у свою чергу взаємно однозначно відповідає набору кольорів для всіх граней: цифри з множини {0, 1, 2, ..., m–1} можна тлумачити як номери кольорів, а кожне місце у записі зв'язати з певною гранню.*

4. Надати змінній n_{out} величину 0.

5. Розглядаючи усі числа від 0 до m^n-1 включно у порядку зростання, при виявленні невикресленого числа:

- збільшити величину n_{out} на 1;
- викреслити виявлене число — найменше число, що відповідає певному способу розфарбування;
- для кожного елемента групи симетрій викреслити натуральне число, запис якого довжини n у системі числення з основою m утворено перестановкою цифр запису викресленого на попередньому кроці числа згідно з правилом перестановки граней — місць запису.

6. Вивести величину n_{out}.

При зростанні m такий алгоритм вимагатиме занадто багато оперативної пам'яті. Спроба зменшити використаний обсяг пам'яті за рахунок збільшення часу обчислень призводить до такої модифікації алгоритму.

1. Зчитавши вхідні дані, встановити кількість граней і відношення суміжності.

2. За допомогою оптимізованого перебору встановити групу симетрій — взаємно однозначних відображень множини граней многогранника у себе.

3. Надати змінній n_{out} величину 0.

4. Розглядаючи усі цілі числа k від 0 до m^n-1 включно у порядку зростання, робити таке:

- для кожного елемента групи симетрій знайти натуральне число, запис якого довжини n у системі числення з основою m утворено перестановкою цифр запису числа k згідно з правилом перестановки номерів граней, що є номерами місць запису, і перевірити, чи знайдене число не менше, ніж k;
- якщо нерівності справджуються для всіх елементів групи симетрій (інакше кажучи, якщо число k є найменшим серед тих, що відповідають певному способу розфарбування), збільшити величину n_{out} на 1.

5. Вивести величину n_{out}.

Але при накладених обмеженнях на час (1 секунда) таке розв'язання набере менше балів — 74 бали.

Рівень 3 (100 балів). Найефективніше *в умовах олімпіади* розв'язання ґрунтується на наслідку теореми Редфілда-Пойя. Теоретичний матеріал українською мовою вичерпно подано за такою адресою: http://kievoi.narod.ru/lectures/polya.html. Алгоритм, на відміну від теорії, легкий для сприйняття і програмного втілення:

1. Зчитавши вхідні дані, встановити кількість граней і відношення суміжності.

2. За допомогою оптимізованого перебору встановити групу симетрій S — взаємно однозначних відображень множини граней многогранника у себе — і кількість елементів цієї групи $|S|$.

3. При всіх k в межах від 1 до $|S|$ змінній p_k надати величину 0.

4. Для кожного елемента групи симетрій:

- знайти j — кількість циклів, на які він розкладається (фрагмент задачі 8.1 відбірково-тренувальних зборів);
- збільшити на 1 величину p_j.

5. Обчислити $n_{\text{out}}=(p_1\cdot m+p_2\cdot m^2+p_3\cdot m^3+\ldots+p_{|S|}\cdot m^{|S|}):|S|$.

6. Вивести величину n_{out}.

Рівень 4 — ідеальне розв'язання, *прийнятне лише за умови істотного збільшення часу виконання*, наприклад для заочного туру. Платонові тіла вивчають у школі. Архімедові тіла давно відомі — див. «Математическая энциклопедия», главный редактор И. М. Виноградов, М.: Советская Энциклопедия, том 3, 1982, стаття *Многогранник*, том 4, 1984, стаття *Полуправильные многогранники*. Або див. http://en.wikipedia.org/wiki/Archimedean_solid чи http://mathworld.wolfram.com/ArchimedeanSolid.html.

Маючи достатній запас часу, можна:

1) обчислити коефіцієнти $\{p_k\}$ тіл, що задовольняють умови завдання;

2) використати у програмі масиви сталих, у яких закладено коефіцієнти $\{p_k\}$ для тіл, що задовольняють умови завдання;

3) передбачити, що програма:
- згідно із вхідними даними визначає вид многогранника за кількістю граней певного вигляду;
- для зчитаної величини m обчислює величину
$$(p_1 \cdot m + p_2 \cdot m^2 + p_3 \cdot m^3 + ... + p_{|S|} \cdot m^{|S|}):|S|,$$
яку і записує у вихідний файл.

Для *тестування* запропоновано 50 тестів:
- m=1 для тестів 1–10;
- m=2 для тестів 11–20;
- m=3 для тестів 21–30;
- m=4 для тестів 31–40;
- m=5 для тестів 41–50.

Залежно від останньої цифри номера тесту розглянуто такі многогранники:
- 1 — тетраедр (4 грані 3-кутники);
- 2 — куб (6 граней 4-кутники);
- 3 — октаедр (8 граней 3-кутники);
- 4 — додекаедр (12 граней 5-кутники);
- 5 — ікосаедр (20 граней 3-кутники);
- 6 — зрізаний тетраедр (4 грані 6-кутники + 4 грані 3-кутники);
- 7 — зрізаний куб (6 граней 8-кутники + 8 граней 3-кутники);
- 8 — зрізаний октаедр (8 граней 6-кутники + 6 граней 4-кутники);
- 9 — кубооктаедр (8 граней 3-кутники + 6 граней 4-кутники);
- 0 — призма (20 граней 4-кутники + 2 грані 20-кутники).

3. Істотні інверсії (автор — Олександр Рибак)

Підрахунок t-істотних інверсій ґрунтується на впорядкуванні послідовності *злиттям*. Процес такого впорядкування складається з кількох кроків.

Спочатку всю послідовність розбивають на послідовності, що містять лише по 1 елементу. Такі частини вже є впорядкованими.

На кожному наступному кроці вже впорядковані частини послідовності групують у пари у порядку зустрічі елементів у початковій послідовності. Якщо якась послідовність не стала елементом пари на цьому кроці, то вона на цьому кроці не зазнає жодних змін. Кожну утворену пару послідовностей перетворюють на одну впорядковану послідовність, переміщуючи до неї (крок за кроком) менший з перших елементів послідовностей пари (приклад такого перетворення послідовностей подано далі таблицею 1).

Утворення й перетворення пар послідовностей здійснюють, поки не отримають лише одну впорядковану послідовність.

Покажемо, як у процесі впорядкування злиттям знайти кількість t-істотних інверсій. Для шуканої кількості запровадимо змінну I, початкова величина якої — 0. У процесі впорядкування пари впорядкованих послідовностей $(a_1, a_2, ..., a_r)$ і $(b_1, b_2, ..., b_s)$ запровадимо змінну q, яку будемо змінювати таким чином. Якщо в деякий момент у другій послідовності залишаться елементи $(b_k, b_{k+1}, ..., b_s)$, то q визначають з умови: a_q — (на даний момент виконання алгоритму) це найперше число першої послідовності, для якого справджується така нерівність: $a_q>b_k+t$. Якщо такого числа не існує, $q=r+1$. Після переміщення b_k до новостворюваної послідовності такі пари: $(a_q, b_k), (a_{q+1}, b_k), ..., (a_r, b_k)$ перестають утворювати t-істотні інверсії (у новостворюваній послідовності відповідні елементи буде розташовано у порядку зростання). Кількість таких пар дорівнює $r+1-q$. Саме на цю кількість одразу після переміщення b_k лічильник кількості інверсій I збільшимо на $r+1-q$, *після* чого поновимо величину q. Завдяки впорядкованості розглядуваних послідовностей, це можна зробити бінарним пошуком, шляхом послідовного збільшення q до справдження сукупності співвідношень: $a_q>b_{k+1}+t$ або $q=r$.

Подамо таблицею 1 послідовні кроки перетворення пари послідовностей 1, 3, 4, 7 та 1, 2, 9 при t=2 і початковій величині I=3 методом злиття.

Зауважимо: I та q змінюють лише тоді, коли переміщують елемент другої послідовності. Коли одна з послідовностей стане порожньою, величини цих змінних залишатимуться сталими.

Оцінимо ефективність алгоритму. Скільки операцій припадає на опрацювання послідовностей з r і s елементів? Щоб вибрати й перемістити елемент до створюваної послідовності, потрібно лише порівняти деякі два елементи й перенести менший з них. Маємо $O(r+s)$ операцій загалом. При оновленні величини q достатньо проглянути першу послідовність лише в напрямку зростання індексу або використати бінарний пошук. Тому загальну кількість змін q оцінюємо відповідно як $O(r)$. Кількість операцій при перетворенні пари послідовностей має порядок $O(r+s)$. Тому для всіх пар на одному кроці потрібно виконати $O(n)$ операцій. Кількість кроків не перевищує $\log_2 n+1$. Отже, загальна кількість операцій становить $O(n \log_2 n)$.

4. Фортеця (автор — Олександр Рибак)

Точки, у яких можна розташовувати вежі, будемо називати *відміченими*. Впорядкуємо ці точки за зростанням першої координати, а при однакових перших координатах — за зростанням другої. Згідно з нашим впорядкуванням занумеруємо відмічені точки натуральними числами від 1 до n, де n – кількість цих точок. Розглянемо довільний невироджений опуклий многокутник — реалізацію розв'язання задачі: всі вершини цього многокутника розташовані у відмічених точках, а його межа містить найбільшу можливу кількість таких точок. Нехай A – вершина многокутника, яка має найменший номер серед усіх його вершин, а B – вершина, яка має найбільший номер.

Введемо деякі означення. Ламану лінію $V_0V_1...V_k$ вважатимемо *опуклою вгору*, якщо при всіх j=1, ..., k−1 промінь, проведений з вершини V_j вертикально вниз, перетинає відрізок $V_{j-1}V_{j+1}$. Аналогічно, ламану $V_0V_1...V_k$ вважатимемо *опуклою вниз*, якщо при всіх j=1, ..., k−1 промінь, проведений з вершини V_j ве-

Таблиця 1

Перша послідовність пари	Друга послідовність пари	Створювана послідовність	I	q
<u>1</u>, 3, 4, 7	1, 2, 9		3	3
3, 4, 7	1, 2, 9	1	3	3
3, 4, 7	2, 9	1, 1	5 = 3 + (4+1 – 3)	4
3, 4, 7	9	1, 1, 2	6 = 5 + (4+1 – 4)	5
4, 7	9	1, 1, 2, 3	6	5
7	9	1, 1, 2, 3, 4	6	5
	9	1, 1, 2, 3, 4, 7	6	5
		1, 1, 2, 3, 4, 7, 9		

ртикально вгору, перетинає відрізок $V_{j-1}V_{j+1}$. У тому випадку, коли ламана є відрізком, вважатимемо її опуклою одночасно вгору і вниз.

Вершини A та B розділяють межу многокутника на дві ламані лінії. За вибором A та B, одна з ламаних опукла вниз, а інша – вгору. Позначимо їх як l_1 та l_2 відповідно. Очевидно, що при русі від A до B вздовж l_1 або l_2 відмічені точки на цих лініях лежать у порядку зростання номерів. Многокутник невироджений, тому можливі лише такі три випадки:

1) l_1 складається з декількох відрізків, а l_2 є відрізком;

2) l_1 є відрізком, а l_2 складається з декількох відрізків;

3) обидві ламані l_1 та l_2 складаються з декількох відрізків.

У кожному з випадків ламані лінії, що складаються з декількох відрізків, мають містити максимальну кількість відмічених точок серед ліній свого типу. Наприклад, у першому випадку l_1 містить максимальну кількість відмічених точок серед усіх ламаних, які сполучають A та B, складаються з декількох відрізків і опуклі вгору. У другому випадку l_2 містить максимальну кількість таких точок серед усіх ламаних, які сполучають A та B, складаються з декількох відрізків і опуклі вниз. Те ж саме для третього випадку. Якщо одна зі згаданих умов не справджується, відповідну ламану можна замінити на лінію того самого типу з більшою кількістю точок. Тоді новий многокутник також буде невиродженим і опуклим, але міститиме більшу кількість відмічених точок. А це суперечить припущенню про максимальність відмічених точок для нашого многокутника. Отже, для кожної пари A, B достатньо знайти, яку найбільшу кількість відмічених точок може містити ламана, що сполучає A та B, складається з декількох відрізків і опукла вгору. Те ж саме для ламаних, опуклих вниз.

Назвемо ламану лінію *правильною*, якщо вона задовольняє такі умови:

1) всі її вершини є відміченими точками;

2) при русі вздовж ламаної від кінця з меншим номером до кінця з більшим номером номери відмічених точок на цій ламаній зростають;

3) ламана опукла вгору;

4) ламана не є відрізком.

Трійку відмічених точок (A, C, B) назвемо *впорядкованою*, якщо A має менший номер, ніж C, а C — менший номер, ніж B. Нехай $F(A, C, B)$ — найбільша можлива кількість відмічених точок, розташованих на *правильній* ламаній, де A є першою, C — передостанньою, а B – останньою відміченою точкою. Якщо відповідної ламаної не існує, покладемо $F(A, C, B)=0$.

Перед тим, як шукати $F(A, C, B)$ для всіх упорядкованих трійок, зробимо декілька загальних спостережень. Впорядковану трійку (A, C, B) називатимемо *правою*, якщо ламана ACB опукла вгору. Якщо при цьому C не лежить на відрізку AB, трійку називатимемо *строго правою*. Нехай $S(C, B)$ – кількість відмічених точок на відрізку CB. Згідно з даними означеннями, на правильній ламаній трійка (A, C, B) має бути строго правою. Тому для інших трійок маємо $F(A, C, B)=0$. Також потрібної ламаної не існує, якщо $S(C, B)>2$ — у цьому випадку C не може бути передостанньою точкою. З іншого боку, якщо (A, C, B) строго права і $S(C, B)=2$, то $F(A, C, B)>0$. Справді, тоді на роль потрібної ламаної підходить об'єднання відрізків AC та CB. У зв'язку з цим строго праві трійки (A, C, B), для яких $S(C, B)=2$, називатимемо *перспективними*.

Перейдемо до підрахунку $F(A, C, B)$. Кожне значення A розглянемо окремо. Для всіх C та B, для яких (A, C, B) є впорядкованою трійкою, знайдемо $F(A, C, B)$ методом динамічного програмування. А саме, за певного значення B нам буде потрібно розрахувати $F(A, C, B)$ для всіх таких C, що (A, C, B) — впорядкована трійка. Ми зробимо це на основі значень $F(A, P, Q)$, розрахованих для всіх таких впорядкованих трійок (A, P, Q), де Q має менший номер, ніж B. Як уже зазначалося, якщо (A, C, B) не перспективна, то $F(A, C, B) = 0$. У випадку перспективної трійки (A, C, B) визначимо $F(A, C, B)$ як максимум з двох значень $\max\{F(A, D, C)+1$, де (D, C, B) — права трійка$\}$ та $S(A, C)+1$. Якщо відповідних правих трійок (D, C, B) немає, покладемо: $F(A, C, B)= S(A, C)+1$. Дійсно, потрібну ламану для трійки (A, C, B) можна отримати, якщо приєднати відрізок CB до відповідної ламаної для (A, D, C) або до відрізка AC. Залишилось обрати той варіант, який дасть лінію з найбільшою кількістю відмічених точок.

Нехай у парі (A, B) точка A має менший номер, ніж B. Для кожної такої пари нехай $U(A, B)=\max\{F(A, C, B)$, де (A, C, B) — впорядкована трійка$\}$. А якщо для даних A та B впорядкованих трійок (A, C, B) немає, покладемо: $U(A, B)=0$. Тоді $U(A, B)$ дорівнює максимальній кількості відмічених точок, що лежать на деякій ламаній з кінцями в A та B, яка опукла вгору і не є відрізком.

Точно так само знайдемо $V(A, B)$ — максимальну кількість відмічених точок, що лежать на деякій ламаній з кінцями в A та B, яка опукла вниз і не є відрізком. Цю задачу можна звести до попередньої, якщо замінити всі координати точок на протилежні за знаком.

Для кожної пари (A, B), де A має менший номер, ніж B, обчислимо: $M(A, B)=\max\{U(A, B)+S(A, B), S(A, B)+V(A, B), U(A, B)+V(A, B)\}-2$.

Величина $M(A, B)$ дорівнює найбільшій кількості відмічених точок, що знаходяться на межі невиродженого опуклого многокутника, у якому вершиною з найменшим номером є A, а з найбільшим — B. Взявши максимум $M(A, B)$ по всіх парах (A, B), отримаємо відповідь на питання задачі.

(Далі буде)

★ ★ ★

На першій сторінці обкладинки: директор Кременчуцького педагогічного училища ім. А.С. Макаренка, Заслужений працівник освіти України Гальченко І.В., викладачі інформатики

Підписано до друку 30.01.2014 р. Формат 60x84 1/8. Папір офсет. Друк офсет. Умовн. друк. арк. 5,88.
Умовн. фарбо-відб. 11,76. Обл.-вид. арк. 8,54. Видавець: ФО–П Жугастрова О.В. Зам. №14–26.
Віддруковано у друкарні видавництва «Фенікс». Свід. ДК 271 від 7.12.2000 р.
Адреса видавця: вул. Половецька, 12/42, к. 88, м. Київ, 04107, Україна.
E-mail: csf221@rambler.ru, www.csf.vashpartner.com.

Передплату на наш журнал можна оформити у будь-якому відділенні зв'язку. Наш індекс 74248