УДК 681.3

МАТЕМАТИЧНІ АСПЕКТИ СТВОРЕННЯ АВТОМАТИЗОВАНОЇ СИСТЕМИ „РЕЄСТР ВИБОРЦІВ УКРАЇНИ”

Ходаков В.Є., Шеховцов А.В., Бараненко Р.В.

Постановка проблеми

В наш час одним із самих актуальних напрямків розвитку сучасної науки є питання проектування інформаційних систем автоматизації управлінської діяльності. У таких системах широке поширення одержала концепція баз даних, відповідно до якої ядром інформаційної системи стають дані, певним чином організовані. Структури організації даних при цьому вибираються у відповідності з багатьма критеріями, серед яких одним з основних є час обробки й пошуку інформації. І тому цілком зрозуміла та увага в літературі, що виявляється до проблем інформаційного пошуку й обробки великих масивів даних [1-7].

Аналіз останніх досліджень

Основна маса праць з дослідження обчислювальної складності алгоритмів пошуку інформації зв'язана з розробкою нових ефективних алгоритмів пошуку, що знаходять численні застосування в різних областях, таких, як машинне проектування, машинна графіка, бібліотечно-інформаційні системи, робототехніка, системи штучного інтелекту і багатьох інших [2-4, 8-12]. У цих роботах оцінюється складність пропонуємих алгоритмів (найчастіше порядок складності) й порівнюється зі складністю раніше розроблених алгоритмів. У ряді праць пропонується інший підхід, зв'язаний із введенням математичних моделей обчислень, що використовуються головним чином для одержання нижніх оцінок складності обчислень [2, 13-17]. Серед цих моделей найбільш відомої є так зване алгебраїчне дерево обчислень Бен-Ора [15]. Як різновид алгебраїчного дерева обчислень можна розглядати алгебраїчне дерево рішень порядку d [17]. У випадку, коли d дорівнює 1, виходить лінійне дерево рішень, з використанням якого отримані докази ряду нижніх оцінок складності [16, 18-20].

Незважаючи на це, задачі оптимальної організації обробки величезних масивів даних й розробки ефективних алгоритмів інформаційного пошуку залишаються актуальними і зараз.

Ціль статті

Метою роботи є визначення математичних аспектів проектування автоматизованої системи „Реєстр Виборців України”, особливостей організації даних у системі, критеріїв, яким має задовольняти автоматизована система, аналіз алгоритмів інформаційного пошуку в системі та розробка моделі залежності часу обробки даних від швидкодії технічних засобів обробки та обсягу даних, що обробляються.

Основний матеріал

Основною функцією автоматизованої системи „Реєстр Виборців України” (АС РВУ) є обробка величезних масивів даних, основу яких складають списки виборців України (близько 37 млн. осіб).

АС РВУ застосовується на рівні оперативного управління виборчим процесом з метою автоматизації управлінської діяльності й служать для рішення задач, що мають високий зміст операцій з обробки даних. До таких операцій відносяться: збір даних, маніпулювання ними, збереження даних і підготовка документів. Маніпуляції з даними виробляються з метою створення з них інформації. До маніпулювання звичайно відносять наступні операції:

 ISBN 7-776-8361-7

- *класифікація* – первинні елементи даних у АС РВУ звичайно мають вигляд кодів, що складаються з одного або декількох символів. Ці коди, що виражають визначені ознаки об'єктів, використовуються для ідентифікації й групування записів;
- *сортування,* що представляє собою процес зміни послідовності записів;
- *обчислення,* що включають у себе арифметичні і логічні операції. Ці операції, що виконуються над елементами даних, дають можливість робити нові елементи даних;
- *укрупнення* – щоб зменшити кількість даних, необхідно їх синтезувати, тобто укрупнювати у формі підсумкових або середніх значень.

Багато даних у АС РВУ необхідно зберігати для наступного використання. Для їхнього збереження створюються спеціальні бази даних.

Існує кілька характеристик, зв'язаних з обробкою даних, що відрізняють АС РВУ від усіх інших комп'ютерних інформаційних систем. У їхньому числі:

- *виконання необхідних задач з обробки даних*;
- *рішення тільки добре структурованих задач,* по яким відомий алгоритм, що веде прямо до обчислення рішення задачі;
- робота в *автоматичному режимі*;
- *використання деталізованих даних.*

Бази даних, що використовуються в АС РВУ, будуються на підставі реляційної моделі [21]. Заслуга розробки і розвитку реляційної моделі баз даних належить Е. Кодду [22-29]. Реляційна база даних складається з плоских таблиць, що називаються *відносинами.* Рядки таблиці (екземпляри записів) називаються *кортежами,* а стовпці — *доменами.*

Для опису відносин і операцій над ними існують точні математичні позначення, засновані на алгебрі відносин або на вирахуванні відносин. У [25] запропонована спеціальна мова маніпулювання даними для такої бази.

Різні користувачі можуть виділяти в базі даних різні набори елементів даних і зв'язки між ними. Отже, необхідно мати можливість витягати підмножини стовпців таблиці для одних користувачів, створюючи таблиці меншої розмірності, а також поєднувати таблиці для інших користувачів, створюючи при цьому таблиці більшої розмірності.

При виборі фізичної організації баз даних вирішальним фактором є ефективність, причому згідно [5] на першому місці стоїть забезпечення ефективності пошуку, далі йдуть ефективність операцій занесення і видалення й потім забезпечення компактності даних.

В теорії дослідження операцій задачі пошуку розуміються як задачі управління зближенням однієї системи (пошукової) з іншою (об'єктом, що шукається) за неповною апріорною інформацією. Розуміється, що мета пошуку — це виявлення об'єкта, що шукається, обумовлене як виконання визначених термінальних умов [30, 31].

Задачею пошуку передбачається багаторазове звернення до тих самих даних, але можливо щоразу з різними вимогами до об'єктів, що шукаються, тобто з різними запитами на пошук. Багаторазове використання породжує особливу проблему — проблему спеціальної організації даних, спрямованої на наступне прискорення пошуку. Процес такої спеціальної організації даних, проведений до того, як здійснюється пошук, називається передобробкою і часто може займати дуже великий час, що потім окупається у результаті багаторазовості пошуку. Найпростішим прикладом передобробки є сортування. Побудова оптимального алгоритму пошуку в цьому випадку зводиться до пошуку оптимальних структур даних, тобто до здійснення такої передобробки даних, що забезпечила б необхідну швидкість пошуку [32].

Для рішення задачі інформаційного пошуку (ЗІП) спочатку необхідно формалізувати саме її поняття [1, 2, 4, 6, 7]. Згідно [32] її формалізація виглядає таким чином:

Нехай нам дані дві безлічі Y і X. Перша безліч Y є безліччю об'єктів пошуку. З елементів цієї безлічі складаються інформаційні масиви, у яких відбувається пошук потрібних об'єктів. Елементи безлічі Y будемо називати *записами.* Друга безліч X назвемо *безліччю запитів,* а його елементи — *запитами.* Нехай на декартовому добутку X x Y задане

бінарне відношення ρ, тобто задані якась підмножина $\boldsymbol{R} \subseteq \boldsymbol{X} \times \boldsymbol{Y}$ і $\boldsymbol{x}\rho\boldsymbol{y}$, якщо $(\boldsymbol{x}, \boldsymbol{y}) \in \boldsymbol{R}$. Відношення ρ будемо називати *відношенням пошуку.* ρ описує критерій семантичної відповідності запису запитові, і будемо говорити, що запис $\boldsymbol{y} \in \boldsymbol{Y}$ *задовольняє* запитові $\boldsymbol{x} \in \boldsymbol{X}$, якщо $\boldsymbol{x}\rho\boldsymbol{y}$.

Трійку $\boldsymbol{S} = \langle \boldsymbol{X}, \boldsymbol{Y}, \rho \rangle$, де X — безліч запитів, Y — безліч записів, ρ — відношення пошуку, задане на X х Y, будемо називати *типом задач інформаційного пошуку.*

Трійку $\boldsymbol{I} = \langle \boldsymbol{X}, \boldsymbol{V}, \rho \rangle$, де X — безліч запитів; V — деяка кінцева підмножина безлічі Y, надалі буде називатися *бібліотекою;* ρ — відношення пошуку, задане на $\boldsymbol{X} \times \boldsymbol{Y}$, будемо називати *задачею інформаційного пошуку* (ЗІП) типу $\boldsymbol{S} = \langle \boldsymbol{X}, \boldsymbol{Y}, \rho \rangle$. Будемо вважати, що завдання $\boldsymbol{I} = \langle \boldsymbol{X}, \boldsymbol{V}, \rho \rangle$ полягає в перерахуванні для довільно взятого запиту $\boldsymbol{x} \in \boldsymbol{X}$ всіх тих і тільки тих записів з V, що знаходяться у відношенні ρ з запитом x, тобто задовольняють запитові x.

Нехай нам дані довільні безлічі запитів X, записів Y і відношення пошуку ρ на X х Y. Причому на безлічі запитів заданий простір імовірностей $\langle \boldsymbol{X}, \sigma, \boldsymbol{P} \rangle$. Наступний результат, що називається тривіальною нижньою оцінкою, справедливий для будь-якої ЗІП при мінімальних обмеженнях. Зміст цього результату полягає в тому, що час пошуку не може бути менше, ніж час, необхідний на перерахування відповіді [32].

Теорема 1 (тривіальна нижня оцінка) *Нехай* $\boldsymbol{I} = \langle \boldsymbol{X}, \boldsymbol{V}, \rho \rangle$ *— довільна ЗІП, F — базова безліч, що задовольняє умові* $U(I, F) \neq \varnothing$, *тоді*

$$T(I, F) \geq \sum_{y \in V} P(O(y, \rho)). \tag{1}$$

Доказ. Візьмемо довільну інформаційну мережу з перемикачами U, що вирішує задачу I. Така мережа існує, тому що $\boldsymbol{U}(\boldsymbol{I}, \boldsymbol{F}) \neq \varnothing$.

Візьмемо довільний запит $\boldsymbol{x} \in \boldsymbol{X}$. Тому що мережа U вирішує ЗІП I, то відповідь на запит x

$$J(x) = \{ y \in V : x\rho y \}. \tag{2}$$

Візьмемо довільний запис $\boldsymbol{y} \in \boldsymbol{J}(\boldsymbol{x})$. Оскільки запис y потрапив у відповідь, то, виходить, у мережі U існує якийсь лист α, якому приписаний запис y і такий, що $\varphi_\alpha(\boldsymbol{x}) = 1$. А тому що $\varphi_\alpha(\boldsymbol{x}) = 1$ і тому що ніякий лист не збігається з коренем, то існує ланцюг, що веде з кореня до листа α, провідність якого дорівнює 1, і в цьому ланцюзі є ребро, що веде в α, із провідністю 1. Це ребро назвемо провідним ребром запису y. Зрозуміло, що різним записам з J відповідають різні провідні ребра, тому що ці ребра ведуть у різні листи. Якщо провідне ребро запису предикатне, предикат, приписаний провідному ребру, обов'язково був обчислений перед тим, як ми потрапили до листа. Якщо провідне ребро запису перемикальне, то обов'язково був обчислений перемикач, приписаний вершині, з якої виходить провідне ребро. Причому такі перемикачі для різних записів з J будуть різними, тому що тільки одне з перемикальних ребер, що виходять з однієї вершини, може мати провідність, рівну 1. У такий спосіб кожному запису з J можна зіставити перемикач або предикат, що обчислюється безпосередньо перед входженням до листа, що відповідає записові. Причому різним записам будуть зіставлені різні перемикачі або предикати. Звідси випливає, що

$$\boldsymbol{T}(\boldsymbol{U}, \boldsymbol{x}) \geq |\boldsymbol{J}(\boldsymbol{x})|. \tag{3}$$

 ISBN 7-776-8361-7

Отже,

$$T(U) = M_x T(U,x) \geq M_x |J(x)| = \int_X |J(x)| P(dx) = \int_X |\{y \in V : x\rho y\}| P(dx) = \sum_{y \in V} \int_{O(y,\rho)} P(dx) = \sum_{y \in V} P(O(y,\rho)). \quad (4)$$

А тому що ця нерівність виконується для будь-якої мережі $U \in \boldsymbol{U}(\boldsymbol{I},\boldsymbol{F})$, то

$$\boldsymbol{T}(\boldsymbol{I},\boldsymbol{F}) \geq \sum_{\boldsymbol{y} \in \boldsymbol{V}} \boldsymbol{P}(\boldsymbol{O}(\boldsymbol{y},\rho)), \quad (5)$$

що і було потрібно довести.

Задача пошуку ідентичних об'єктів складається з пошуку в інформаційному масиві об'єкта, ідентичного об'єктові-запитові.

У нашому випадку необхідно знайти в кінцевій безлічі оброблюваних файлів однакові записи, що містять кінцеву підмножину ідентичних атрибутів із усієї безлічі атрибутів записів у файлах, записати їх до файла, що містить результат пошуку, видаливши ці записи з вхідних файлів, тим самим зменшуючи їхній розмір. Дана задача ускладнюється необхідністю обробки дуже великих масивів даних у кожному файлі (порядком 7 Гбайт), а оскільки кількість оброблюваних файлів дорівнює 3, то загальний масив інформації для первинної обробки збільшується до 21 Гбайт, що вимагає величезних технічних, програмних і часових ресурсів. Нехай

$X_i \in X$ - оброблюваний файл (база даних), що належить кінцевій безлічі оброблюваних файлів (баз даних), де $\boldsymbol{i} \in \{1; \boldsymbol{I}\}$, $\boldsymbol{I}$ - кількість оброблюваних файлів (баз даних);

$n_k \in n$ - запис в оброблюваному файлі (кортеж бази даних), що належить кінцевій безлічі записів в оброблюваному файлі, де $\boldsymbol{k} \in \{1; \boldsymbol{K}\}$, $\boldsymbol{K}$ - кількість записів в оброблюваному файлі (кількість кортежів у базі даних);

$a_j \in a$ - атрибут запису в оброблюваному файлі (домен бази даних), що належить кінцевій безлічі атрибутів запису в оброблюваному файлі, де $\boldsymbol{j} \in \{1; \boldsymbol{J}\}$, $\boldsymbol{J}$ - кількість атрибутів запису в оброблюваному файлі (кількість доменів у базі даних).

Введемо критерій відмінності $0 \leq \delta_{i_1 i_2 k_1 k_2 j} \leq 1$ вмісту атрибутів записів у файлах бази даних друг від друга $\delta_{i_1 i_2 k_1 k_2 j} = a_{i_1 k_1 j} - a_{i_2 k_2 j}$, де $i_1, i_2 \in \{1; I\}; k_1, k_2 \in \{1; K\}$, такий, що

$$\exists \begin{matrix} i \in \{1; I\} \\ k \in (1; K) \end{matrix} \quad \sum_{j=1}^{J} \delta_{ikj} \to 0, \quad mo \quad P(n_{ik}) \to 1, \quad (6)$$

де $P(n_{ik})$ - імовірність того, що в $\boldsymbol{i}$-х файлах $\boldsymbol{k}$-і записи збігаються.

Для вибірки записів до файла, що містить результати пошуку, необхідно ввести критерій вибірки $0 \leq \eta_{ik} \leq 1 \quad \forall i \in \{1; I\}, k \in \{1; K\}$, рівний імовірності збігу $\boldsymbol{k}$-х записів у $\boldsymbol{i}$-х файлах.

При $\eta_{ik} \approx 1$ k-ий запис можна зберегти в файлі, що містить результати пошуку, і видалити його з усіх $\boldsymbol{i}$-х файлів.

Час обробки масиву файлів складає:

$$T = \sum_{i=1}^{I} T_i, \quad (7)$$

де T_i - час обробки $\boldsymbol{i}$-го файлу.

$$T_i = \sum_{k=1}^{R_i \in \{1; K\}} t_{ik}, \quad (8)$$

де t_{ik} - час обробки k-го запису i-го файлу.

У свою чергу

$$t_{ik} = \sum_{j=1}^{J} t_{ikj}, \tag{9}$$

де t_{ikj} - час обробки j-го атрибута k-го запису i-го файлу.

$$t_{ikj} = q_{mikj} * t_m, \tag{10}$$

де q_{mikj} - кількість тактів процесора, що відповідає виконанню операції обробки j-го атрибута k-го запису i-го файлу, t_m - час виконання одного такту процесора ЕОМ, на якій відбувається обробка даних.

Доведено [33-37], що

$$t_m = \frac{1}{f_m}, \tag{11}$$

де f_m - тактова частота процесора ЕОМ.

Таким чином час виконання операцій маніпулювання даними в СЕОД залежить від тактової частоти процесора ЕОМ, на якій відбуваються ці операції, і складає:

$$T = \sum_{i=1}^{I} \sum_{k=1}^{R_i \in \{1;K\}} \sum_{j=1}^{J} \frac{q_{mikj}}{f_m}. \tag{12}$$

Максимальний час пошуку ідентичних об'єктів у нашому випадку обчислюється за формулою (13):

$$T_{max} = \frac{1}{f_m} \cdot I \cdot K \cdot J \cdot q_m. \tag{13}$$

Таким чином для обробки I файлів довжиною $R_i \in \{1;K\} * J$ знадобиться час $T \le T_{max}$, що залежить від тактової частоти процесора ЕОМ і кількості елементів даних для обробки. Чим більше тактова частота процесора і чим менше елементів даних для обробки, тим менше часу займе обробка масиву файлів.

Висновки

Авторами розглянуті математичні аспекти проектування автоматизованої системи „Реєстр Виборців України”, особливості організації даних у системі, перелічені критерії, яким має задовольняти автоматизована система, проаналізовані алгоритми інформаційного пошуку в системі та запропонована модель залежності часу обробки даних від швидкодії технічних засобів обробки та обсягу даних, що обробляються.

In the given article the mathematical aspects of designing of the automated system "The Register of the voters of Ukraine", the features of data structure in this system, criterions which the projected automated system should satisfy are considered, the algorithms of information search in system are analyzed and the model of dependence of data processing time from productivity of technical means of processing and volume of the processable data is proposed.

 ISBN 7-776-8361-7

1. Альсведе Р., Вегенер И. Задачи поиска. - М.: Мир, 1982.

2. Ахо А., Хопкрофт Дж., Ульман Дж. Построение и анализ вычислительных алгоритмов. - М.: Мир, 1979.

3. Кнут Д. Искусство программирования для ЭВМ: Т.3: Сортировка и поиск. - М.: Мир, 1978.

4. Ли Д., Препарата Ф. Вычислительная геометрия. Обзор // Кибернетический сб. -1987. Вып. 24. - С. 5 - 96.

5. Мартин Дж. Организация баз данных в вычислительных системах. - М.: Мир, 1980.

6. Решетников В. Н. Алгебраическая теория информационного поиска // Программирование. - 1979. - № 3. - С. 68 - 74.

7. Селтон Г. Автоматическая обработка, хранение и поиск информации. - М.: Советское радио, 1973.

8. Ньюмен У. М., Спруэлл Р. Ф. Основы интерактивной машинной графики. - М.: Мир, 1976.

9. Солтон Дж. Динамические библиотечно-информационные системы. - М.: Мир, 1979.

10. Chazelle B. M. Filtering search: a new approach to query-answering // Proc. 24th IEEE Annu. Symp. Found. Comput. Sci. - Nov. 1983. - P. 122-132.

11. Edelsbrunner H., Overmars M. H., Siedel R. Some methods ofcomputational-geometry applied to computer graphics // IIG, Technische Univ. Graz, Austrua, Tech. Rep. F117. - June 1983.

12. Lee D. T., Wong C. K. Quintari trees: A file structures for multidimensional database system // ACM Trans. Database Syst. - Sept. 1980. - V. 1, №1. - P. 339-353.

13. Гасанов Э. Э. Некоторые оценки сложности поиска информации // Физическое и математическое моделирование дискретных систем. Межвузовский сборник трудов №56. - М.: Изд-во Моск. энерг. ин-та, 1985. - С. 43-47.

14. Гасанов Э. Э. О виде оптимальных информационных сетей для отношений линейного квазипорядка. Препринт Р-5-303 ИЯФ АН УзССР. - Ташкент, 1987.

15. Ben-Or M. Lower bounds for algebraic computation trees // Proc. 15th ACM Annu. Symp. Theory Comput. - Apr. 1983. - P. 80-86.

16. Dobkin D. P., Lipton R. J. On the complexsity of computations under varying sets of primitives // J. Comput. Syst. Sci. - 1979. - V. 18. - P. 86-91.

17. Steele J. M., Yao A. C. Lower bounds for algebraic decision trees // J. Algorith. - 1982.

18. Dobkin D. P. A nonlinear lower bound on search tree programs for solving knapsack problems // J. Comput. Syst. Sci. - 1976. - V. 13. - P. 69-73.

19. Dobkin D. P., Lipton R. J. A lower bound of $l/2n^2$ on linear search programs for the knapsack problem // J. Comput. Sci. - 1978. - V. 16. - P. 413-417.

20. Yao A. C., Rivest R. L. On the polyhedral decision problem // SIAM J. Comput. - 1980.

21. Праг Керри Н., Ирвин Майкл Р. Access 2000. Библия пользователя.: пер. с англ. – М.: Издательский дом «Вильямс», 2001. – 1040 с. + 32 с. краткого справочника: ил. – Парал. тит. англ.

22. Codd E. F. A Relation Model of Data for Large Shared Data Banks // Comm. ACM 13, №6, ACM, New York, London, Amsterdam, June 1970. P. 377-387.

23. Codd E. F. Further Normalization of the Data Base Relational Model // Courant Computer Sci. Symposia (vol. 6: "Data-Base System"), ed. by R. Rustin, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1972.

24. Codd E. F. Relational Completeness ofData Base Sublanguages // Courant Computer Sci. Symposia (vol. 6: "Data-Base System"), ed. by R. Rustin, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1972.

25. Codd E. F. A Data Base Sublanguage Founded on the Relational Calculus // Proc. of the 1971 ACM-SIGFIDET Workshop on Data Description, Access, and Control, ACM, New York, London, Amsterdam, 1972.

26. Codd E. F. Access Control for Relational Data Base Systems // BCS Symposium on Relational Data-Base Concepts, Apr. 1973, British Computer Soc., London, 1973.

27. Codd E. F. Recent Investigations in Relational Data-Base Systems // Information Processing'74, North-Holland, Amsterdam, 1974.

28. Codd E. F. Relational Database: A Practical Foundation for Productivity // Commun. of ACM. - 1982. - V. 25, №2. P. 140-155.

29. Date C. J., Codd E. F. The Relational and Network Approaches: Comparison of the Application Programming Interfaces // Proc. of the 1974 ACM-SIGFIDET Workshop, ACM, New York, London, Amsterdam, 1974.

30. Зайченко Ю.П. Исследование операций. – К.: Выща школа, 1986.

31. Горелик В.А., Ушаков И.А. Исследование операций. – М.: Машиностроение, 1986. – 288 с.

32. Гасанов Э.Э. Функционально-сетевые базы данных и сверхбыстрые алгоритмы поиска. Конспект лекций. – М.: Издательский центр РГГУ, 1997. – 88 с.

33. Фритч В. Применение микропроцессоров в системах управления: Пер. с нем. – М.: Мир, 1984. - 464 с., ил.

34. Шоу А. Логическое проектирование операционных систем /Пер. с англ. В.В. Макарова и В.Д. Никитина. – М.: Мир, 1981. – 360 с.

35. Цикритзис Д., Бернстайн Ф. Операционные системы /Пер. с англ. В.Л. Ушковой и Н.Б. Фейгельсон. – М.: Мир, 1977. – 336 с.

36. Системное программное обеспечение /А.В. Гордеев, А.Ю. Молчанов. – СПб.: Питер, 2001. – 736 с.: ил.,

37. Столингс Вильям. Операционные системы, 4-е издание.: пер. с англ. – М.: Издательский дом «Вильямс», 2002. – 848 с.: ил. – Парал. тит. англ.

 ISBN 7-776-8361-7