Теорія алгоритмів

Практичне завдання №5

“Хеш-таблиці”

Хеш-таблиці (*hash-tables*) можуть використовуватись для збереження масивів даних, швидкого доступу, вставки та видалення елементів. За допомогою хеш-таблиць можна ефективно розв'язати наступну задачу. Нехай заданий масив чисел A та число S. Потрібно дізнатись, чи присутні в масиві A два числа, сума яких дорівнює S.

Тривіальний алгоритм, який попарно перебере всі можливі пари чисел, працює, відповідно $O(n^2)$ часу, де n — кількість елементів в масиві A. Проте можна значно пришвидшити розв'язок цієї задачі, скориставшись хеш-таблицями.

Алгоритм буде виглядати наступним чином:

1. Додати до хеш-таблиці T всі елементи масиву A. Час роботи цієї операції — $O(n)$.
2. Для кожного числа x з A визначити, чи присутнє в хеш-таблиці T число $S - x$. Якщо так, то ми знайшли два числа, які утворюють цю суму та знаходяться у вхідному масиві A. Час роботи цього етапу — $O(n)$.

Таким чином, загальний час розв'язку задачі із використанням хеш-таблиць становить $O(n)$.

Завдання

В роботі необхідно реалізувати різні типи хеш-таблиць із використанням різних хеш-функцій для розв’язання наведеної вище задачі. При цьому потрібно порівняти ефективність різних підходів шляхом підрахунку кількості колізій для кожного типу хеш-функцій та хеш-таблиць.

Нижче наведений перелік типів хеш-таблиць та хеш-функцій:

1. Хеш-таблиця **на основі ланцюгів** (*chained hash*) із використанням хеш-функції **за методом ділення**.
2. Хеш-таблиця **на основі ланцюгів** (*chained hash*) із використанням хеш-функції **за методом множення**.

Завдання на додаткові бали

3. Хеш-таблиця **на основі відкритої адресації** (*open address hash*) із використанням хеш-функції **за методом лінійного дослідження**.
4. Хеш-таблиця **на основі відкритої адресації** (*open address hash*) із використанням хеш-функції **за методом квадратичного дослідження**.
5. Хеш-таблиця **на основі відкритої адресації** (*open address hash*) із використанням хеш-функції **за методом подвійного хешування**.

Увага!! Розмір хеш-таблиці T не повинен перевищувати потрійного розміру вхідного масиву A.

Формат вхідних/вихідних даних

Розроблена програма повинна зчитувати вхідні дані з файлу заданого формату та записувати дані у файл заданого формату. У вхідному файлі зберігається масив чисел A та масив сум S.

Вхідний файл представляє собою текстовий файл із $N+M+1$ рядків. Перший рядок містить два числа: N та M, де N — кількість елементів в масиві A, M — кількість сум S. Далі йде відповідно N рядків, кожен з яких містить один елемент масиву A. Після йде ще M рядків, кожен з яких містить одне число — суму S_i.

Після зчитування вхідних даних, програма визначає, чи є в масиві A два числа x та y такі, що $x+y=S_i$, при цьому використовуючи тип хеш-таблиці та хеш-функції під номером k (від 1 до 5). Номер k передається як другий аргумент командного рядку.

Вихідний файл представляє також текстовий файл з $M+1$ рядків. Перший рядок вихідного файлу містить кількість підрахованих колізій для типу хеш-таблиці k. Далі йде M рядків, кожен з яких містить два числа — доданки x та y для суми S_i з вхідного файлу (i від 1 до M). Якщо у вхідному масиві A немає двох елементів x та y таких, що $x+y=S_i$, то відповідний рядок у вихідному файлі повинен містити два нулі: “0 0”.

Увага! Кількість підрахованих колізій у вашій реалізації хеш-таблиці не повинна відрізнятись від кількості колізій у відповідному вихідному еталонному файлі більше аніж у два рази.

До документу завдання також додаються приклади вхідних і вихідних файлів різної розмірності. Нижче наведені приклади вхідного та вихідного файлу для $N = 10$ та $M = 5$.

Вхідний файл	Вихідний файл
10 5 40 12 79 35 43 52 83 66 89 79 88 87 126 78 122	1 0 0 35 52 43 83 12 66 79 43

Вимоги до програмного забезпечення

- Розробляти програму можна на одній з наступних мов програмування: **C/C++** (версія C++11)**, C#** (версія C# 5.0)**, Java** (версія Java SE 8)**, Python** (версія 2.7).
- Програма повинна розміщуватись в окремому висхідному файлі, без використання додаткових нестандартних зовнішніх модулів.
- Не дозволяється використовувати будь-які нестандартні бібліотеки та розширення. Програма не повинна залежати від операційної системи. Якщо ви не впевнені відносно того, які бібліотеки відносяться до стандартних, зверніться до викладача.
- Не реалізуйте жодного інтерфейсу користувача (окрім командного рядку). Програма не повинна запитувати через пристрій вводу в користувача жодної додаткової інформації. Вашу програму будуть використовувати виключно у вигляді “чорного ящику”.
- **Назва висхідного файлу** вашої програми повинна задовольняти наступному формату: *НомерГрупи_ПрізвищеСтудента_НомерЗавдання.Розширення*, де *НомерГрупи* — це один з рядків is31, is32, is33; *ПрізвищеСтудента* — прізвище студента записане латинськими літерами; *НомерЗавдання* — двозначний номер завдання (01, 02, ...); *Розширення* — розширення файлу, відповідно до мови програмування (.c, .cpp, .cs, .java, .py). Приклад назви висхідного файлу: *is31_ivanenko_03.cs*.
- Розроблена програма повинна зчитувати з командного рядку назву вхідного файлу та записувати результат у вихідний файл. При запуску першим і **єдиним аргументом**

командного рядку повинна бути назва вхідного файлу (наприклад, *input_10.txt*). Назва вихідного файлу повинна складатись із назви файлу самої програми разом із суфіксом "*_output*" і мати розширення .txt. Приклад назви вихідного файлу: *is31_ivanenko_03_output.txt*.

- Якщо ви не реалізували заданий тип хеш-таблиці, то програма не повинна нічого виконувати, а відразу припинити свою роботу (у вихідний файл нічого не записується).

Нараховані бали

За успішне виконання даного завдання нараховується **5 балів**. Якщо реалізовані типи хеш-таблиць на додаткові бали (типи №3-5), то додаються ще **2 бали**.

Також за ранню здачу нараховується один додатковий бал. Але, якщо на момент ранньої здачі програма не працює коректно, то додаткові бали не нараховуються.

Можна отримати ще один додатковий бал за ефективне алгоритмічне рішення: коли алгоритм є компактним та елегантним. Цей бал враховується тільки у випадку вчасної здачі програми (коли ще не нараховуються штрафні бали).

Термін здачі

Практичне завдання №5 необхідно здати до **23:59 29 березня 2014 року**.

Якщо ви завантажили робочу версію програми до 23:59 22 березня, то це вважається ранньою здачею, за яку нараховуються додаткові бали.

Якщо ви не встигли надати вашу програму до вказаного часу, то маєте додатковий час до 23:59 12 квітня, але в такому разі буде нараховано тільки 50% балів.

Як завантажувати відповідь на завдання

Розроблена програма у вигляді висхідного файлу із відповідною назвою завантажується в систему Moodle (http://moodle.asu.ntu-kpi.kiev.ua/) в курс "**Теорія алгоритмів (КН)**". Файл програми необхідно завантажити у розділ "**Практичне завдання №5**". Дозволяється робити декілька спроб завантаження. У випадку кількох завантажень оцінюватись буде тільки останній завантажений варіант.

Рекомендована література

Конспект лекцій з курсу "Теорія алгоритмів": **Лекція 08 — Хеш-таблиці**