

Київський національний університет імені Тараса Шевченка

Факультет комп'ютерних наук та кібернетики

Кафедра інтелектуальних програмних систем

Алгоритми та складність

Завдання №

«Узагальніть метод Рабіна-Карпа пошуку зразка в текстовому рядку так, щоб він дозволив розв'язати задачу пошуку заданого зразка розміром m на m у символьному масиві розміром n на n . Зразок можна рухати по горизонталі та вертикалі, але не обертати.»

Варіант №3

Виконала студентка 2-го курсу

Групи ІПС-22

Клевчук Марія Вячеславівна

Київ - 2024

Завдання

Розробити програму на мові C++, що реалізує узагальнення методу Рабіна-Карпа для пошуку шаблону розміру m на m в двовимірному масиві розміру n на n . Забезпечити можливість пошуку зразка по горизонталі та вертикалі без його обертання. Програма повинна повертати координати верхнього лівого кута кожного знайденого входження зразка.

Теорія

Метод Рабіна-Карпа для одновимірного пошуку ґрунтується на обчисленні хеш-значення шаблону та порівнянні його з хеш-значеннями підрядків в тексті. Якщо хеш-значення збігається, проводиться перевірка кожного символу для підтвердження збігу.

Узагальнення для двовимірного пошуку

В узагальненому методі Рабіна-Карпа обчислюється хеш значення для двовимірного підмасиву розміру m на m у вихідному масиві розміру n на n . За рахунок використання переобчислення хеша (re-hashing) після кожного зсуву шаблону, алгоритм забезпечує ефективність порівнянь.

Хеш-значення — це результат обчислення хеш-функції, яка перетворює вхідні дані (наприклад, рядок символів, файл або ціле число) на короткий, фіксований розмір значення (зазвичай це число або набір байтів). Хеш-функція призначена для того, щоб швидко та ефективно відобразити великі обсяги даних у компактне представлення.

Алгоритм

- 1) Зразок розміром $m*m$ подається на вхід і для нього обчислюється хеш-значення.
- 2) Для кожного рядка текстового масиву розміром $n*n$ обчислюються хеші підрядків довжини $m*m$ по горизонталі. Це дозволяє зберігати хеші для всіх рядків, які згодом будуть об'єднані по вертикалі.
- 3) Порівняння хешів і підтвердження збігу: алгоритм послідовно переміщає "вікно" розміром $m*m$ по всьому текстовому масиву, обчислюючи хеш-значення кожного підмасиву. Якщо хеш підмасиву збігається з хешем зразка, проводиться детальна перевірка символів. Якщо всі символи підмасиву

збігаються зі зразком, координати верхнього лівого кута цього входження записуються як позиція знайденого збігу.

- 4) Якщо зразок знайдено в текстовому масиві, алгоритм повертає позиції всіх його входжень. Якщо збігів немає, алгоритм повідомляє про відсутність збігів.

Складність алгоритму

Обчислення хешів для першого підмасиву і зразка складає $O(n^2)$.

Проходження по всіх підмасивах тексту складає $O((n-m+1)^2)$

Якщо припустити, що значення m та n доволі близькі, то складність можна обрахувати за формулою $O(n^2 \cdot m^2)$

Мова реалізації алгоритму

C++

Модулі програми

void readGrid(std::vector<std::string>& grid, int size, const std::string& name)

- функція читає сітку символів розміром $m \cdot m$ або $n \cdot n$ із введення.

void precomputePowers(std::vector<int64_t>& pow_base, std::vector<int64_t>& pow_base_row, int max_len) - функція обчислює степені для основ хешування.

int64_t computePatternHash(const std::vector<std::string>& pattern, int m) - функція обчислює хеш патерну.

void computeTextRowHashes(const std::vector<std::string>& text, std::vector<std::vector<int64_t>>& text_row_hashes, int n, int m, const std::vector<int64_t>& pow_base) - функція обчислює хеші для кожного рядка тексту

void searchPattern(const std::vector<std::string>& text, const std::vector<std::string>& pattern, int n, int m, const std::vector<std::vector<int64_t>>& text_row_hashes, int64_t pattern_hash, const std::vector<int64_t>& pow_base_row) - функція виконує основну логіку пошуку патерну з перевіркою хешів та порівнянням символів.

Інтерфейс користувача

Для вводу вхідних даних алгоритм використовує консоль.

Вхідні дані:

- Значення n
- Значення m
- Символьні рядки масиву
- Зразок, який шукається в символьних рядках

Вихідні дані:

- Якщо зразок не було знайдено в символьних рядках, в консоль виводиться повідомлення про це
- Якщо зразок було знайдено в символьних рядках, в консоль виводиться

Тестові приклади

Приклад 1

```
Enter the size of the text grid (n): 5
Enter the text grid:
qwert
yuiop
asdfg
hijklz
xcvbn
Enter the size of the pattern grid (m): 2
Enter the pattern grid:
qw
yu
Pattern found at position (0, 0)
Program ended with exit code: 0
```

```
Enter the size of the text grid (n): 3
Enter the text grid:
aaa
ddd
fff
Enter the size of the pattern grid (m): 2
Enter the pattern grid:
aa
bb
Pattern not found in the text grid.
Program ended with exit code: 0
```

Приклад 2

Висновки

Реалізація узагальненого методу Рабіна-Карпа дозволяє ефективно здійснювати пошук шаблону в двовимірному масиві. За рахунок використання переобчислення хешів алгоритм забезпечує швидке порівняння підмасивів, що дозволяє уникнути зайвих обчислень.

Використані літературні джерела

<https://www.geeksforgeeks.org/rabin-karp-algorithm-for-pattern-searching-in-matrix/>

https://uk.wikipedia.org/wiki/Алгоритм_Рабіна_—_Карпа

https://medium.com/@gnaneshwarreddy_alla/rabin-karp-algorithm-for-pattern-searching-480872ecb846