Київський національний університет імені Тараса Шевченка

Факультет комп`ютерних наук та кібернетики

Кафедра інтелектуальних інформаційних систем

Алгоритми та складність

Завдання №7

Виконав студент 2-го курсу

Групи К-28

Самойлич Євгеній Костянтинович

2020

**Завдання**

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

**Теорія**

Алгоритми пошуку зразка в текстовому рядку повертають номер першого символу, після якого послідовність символів співпала, інакше -1. У даному ж випадку алгоритм буде повертати пару чисел: номер рядка та стовбця верхнього лівого символу шаблону в тексті, або ж пару {-1, -1}.

**Алгоритм**

В алгоритмі Рабіна-Карпа створюється хеш шаблону, що шукається, і перевіряється, чи хеш підрядку тексту, в якому шукаємо, відповідає шаблону. Якщо не відповідає, то шаблон не може бути присутній в тексті. Якщо відповідає, то посимвольно перевіряємо.

Нехай дано рядок s[0...n−1]. Тоді поліноміальним хешем s називається число , де p — деяке просте число, а s[i] − код i-ого символу рядка s. Знаючи хеш s[i,…, j] можна обрахувати хеш s[i+1,…, j+1] наступним чином: *hash(s[i+1,…, j+1]) = (1/p) × ( hash(s[i,…, j]) – s[i] ) + s[j+1] × pn-1*, де n – довжина шаблону.

**Алгоритм Рабіна-Карпа**: підраховуємо хеш для шаблона. Далі, рухаючись зліва направо, обраховуємо хеші підрядків основного рядку, використовуючи значення хешу для підрядку, зміщеного на 1 ліворуч, аби проводити менше операцій. Якщо цей хеш співпадає з хешем шаблону, проводимо посимвольну перевірку, інакше – рухаємося далі.

**Алгоритм нашої задачі:** Нехай дано Text[m][m] та шаблон s[n][n]. Для виконання даної задачі будемо шукати в і-ому рядку тексту (0<=i<=n-1) 0-ий рядок шаблону алгоритмом Рабіна-Карпа. При її знаходженні перевіряємо всі наступні відрізки під підрядком алгоритмом Рабіна-Карпа.

**Складність**

В найгіршому випадку складність алгоритму Θ((M-N+1)×N), в середньому та найкращому випадках O(M+N), де M – розмір масиву тексту, N – розмір масиву шаблону.

**Мова програмування**

С++

**Модулі програми**

* std::pair<int, int> RK\_matrix(const std::vector<std::string>& text, const std::vector<str::string>& pattern)

Приймає масиви рядків тексту та масиви рядків шаблону.

**Інтерфейс користувача**

Вхідні дані вводяться користувачем у консоль. Результат виводиться користувачу у консоль.

**Тестові приклади**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Text** | | | | |
| **a** | **r** | **r** | **s** | **d** |
| **a** | **a** | **r** | **r** | **q** |
| **z** | **z** | **x** | **c** |  |
| **y** | **y** | **u** | **p** | **t** |
| **l** | **m** | **n** | **o** | **p** |

|  |  |  |
| --- | --- | --- |
| **Pattern** | | |
| **a** | **r** | **r** |
| **z** | **x** | **c** |
| **y** | **u** | **p** |

Спочатку відбувається пошук входження першого рядка шаблона в тексті.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Text** | | | | |
| **a** | **r** | **r** | **s** | **d** |
| **a** | **a** | **r** | **r** | **q** |
| **z** | **z** | **x** | **c** | **w** |
| **y** | **y** | **u** | **p** | **t** |
| **l** | **m** | **n** | **o** | **p** |

|  |  |  |
| --- | --- | --- |
| **Pattern** | | |
| **a** | **r** | **r** |
| **z** | **x** | **c** |
| **y** | **u** | **p** |

Далі, якщо знайдено збіг, перевіряються підрядки тексту під збігом. Збігів не знайдено.

Пошук першого рядка в тексті продовжується. Більше у першому рядку тексту немає збігів з першим рядком шаблону.

Порівнюємо другий рядок тексту та перший рядки шаблону. Є збіг

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Text** | | | | |
| **a** | **r** | **r** | **s** | **d** |
| **a** | **a** | **r** | **r** | **q** |
| **z** | **z** | **x** | **c** | **w** |
| **y** | **y** | **u** | **p** | **t** |
| **l** | **m** | **n** | **o** | **p** |

|  |  |  |
| --- | --- | --- |
| **Pattern** | | |
| **a** | **r** | **r** |
| **z** | **x** | **c** |
| **y** | **u** | **p** |

Порівнюємо далі підрядки тескту та шаблону, є збіги. Усі рядки шаблону співпали з підрядками тексту алгоритм повернув {1, 1}.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Text** | | | | |  |
| **a** | **r** | **r** | **s** | **d** | **Pattern** | | |
| **a** | **a** | **r** | **r** | **q** | **a** | **r** | **r** |
| **z** | **z** | **x** | **c** | **w** | **z** | **x** | **c** |
| **y** | **y** | **u** | **p** | **t** | **y** | **u** | **p** |
| **l** | **m** | **n** | **o** | **p** |

**Висновки**

Модифікація алгоритму Рабіна-Карпа для пошуку входжень матричного шаблону в матричний текст доволі очевидно випливає з базової реалізації алгоритму. Час роботи реалізованого алгоритму в середньому та кращому випадках становить O(M+N).

**Література**

* Лекція 8
* https://www.geeksforgeeks.org/rabin-karp-algorithm-for-pattern-searching-in-matrix/
* https://e-maxx.ru/algo/rabin\_karp