**Київський національний університет імені Тараса Шевченка**

**Факультет комп'ютерних наук та кібернетики**

**Алгоритми і складність**

**Лабораторна робота 7**

**Звіт**

**Підготував:**

студент групи К-29

Григорович Олег Андрійович

**Київ-2019**

**1. Постановка завдання**

Узагальнити метод Рабіна-Карпа пошуку зразка в текстовому рядку так, щоб він дозволив вирішити завдання пошуку заданого зразка розміром m на m в символьному масиві розміром n на n. Зразок можна рухати по горизонталі і вертикалі, але не обертати.

**2. Опис алгоритму**

Нехай є текст Text [n] [n] і шаблон s [m] [m]. Для виконання даного завдання будемо слідувати таким чином. Будемо шукати в i-му рядку, де (0 <= i <= n-1), 0-ву рядок s Алгоритмом Рабіна-Карпа. При знаходженні її - перевіряємо відрізки рядки під нею все тим же алгоритмом Рабіна-Карпа на відповідність.

**3. Алгоритм Рабіна-Карпа**

Алгоритм починається з підрахунку hash (s [0..m-1]) і hash (p [0..m-1]), а також з підрахунку pm, для прискорення відповідей на запит.

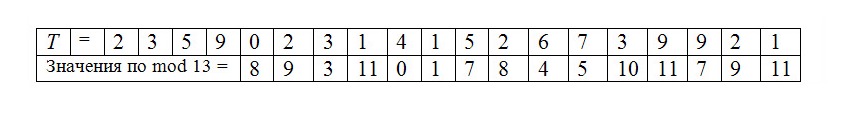
Для i∈ [0..n-m] обчислюється hash (s [i..i + m-1]) і порівнюється з hash (p [0..m-1]). Якщо вони виявилися рівними, то зразок p швидше за все міститься в рядку s починаючи з позиції i, хоча можливі і помилкові спрацьовування алгоритму. Якщо потрібно звести такі спрацьовування до мінімуму або виключити зовсім, то застосовують порівняння деяких символів з цих рядків, які обрані випадковим чином, або застосовують явне порівняння рядків, як в наївному алгоритмі пошуку підрядка в рядку. У першому випадку перевірка відбудеться швидше, але ймовірність помилкового спрацьовування, хоч і невелика, залишиться. У другому випадку перевірка займе час, що дорівнює довжині зразка, але повністю виключить можливість помилкового спрацьовування.

Якщо потрібно знайти індекси входження декількох зразків, або порівняти два рядки - вигідніше буде заздалегіть порахувати всі степені p, а також хеші всіх префіксів рядки s.

**4. Приклад**

Нехай алфавіт D = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}, тобто кожен символ в алфавіті є d-кова цифра, де d = │D│.

Нехай зразок має вигляд W = 3 1 4 1 5

Обчислюємо значення чисел з вікна довжини | W | = 5 по mod q, q - просте число. 23590 (mod 13) = 8, 35902 (mod 13) = 9, 59023 (mod 13) = 9, ...

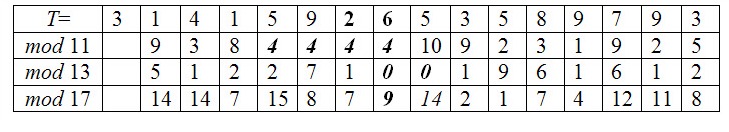
k1 = 314157 (mod 13) - входження зразка,

k2 = 673997 (mod 13) - холосте спрацьовування.

З рівності ki = kj (mod q) не слідує, що ki = kj (наприклад, 31415 = 67399 (mod 13), але це не означає, що 31415 = 67399). Якщо ki = kj (mod q), то ще треба перевірити, чи збігаються рядки W [1 ... m] і T [s + 1 ... s + m] насправді.

Якщо просте число q досить велике, то додаткові витрати на аналіз холостих спрацьовувань будуть невеликі.

У гіршому випадку час роботи алгоритму РК - Θ ((N-M + 1) \* M), в середньому ж він працює досить швидко - за час О (N + M).

Приклад: Скільки холостих спрацьовувань k зробить алгоритм РК, якщо q = 11, 13, 17. Нехай W = {2 6} 

26 mod 11 = 4 → k = 3 холості спрацьовування,

26 mod 13 = 0 → k = 1 холосте спрацьовування,

26 mod 17 = 9 → k = 0 холостих спрацьовувань.

Очевидно, що кількість холостих спрацьовувань k є функцією від величини простого числа q (якщо функція обробки зразка mod q) і, в загальному випадку, від виду функції для обробки зразка W і тексту Т.