**Київський національний університет імені Тараса Шевченка**

**факультет комп’ютерних наук та кібернетики**

**кафедра інформаційних систем**

**Лабораторна робота № 7**

**“Узагальнення методу Рабіна-Карпа”**

**Виконав студент 2 курсу**

**групи К-28**

**Гоженков Єгор Геннадійович**

**2018**

1. **Зміст 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.

1. **Спосіб розв’язання**

Оскільки необхідно знайти підматрицю висотою m, то можна схешувати стовпчики шуканої матриці, а потім зробити теж саме для стовпчиків в перших m рядках великої матриці. Тепер можна застосовувати метод Рабіна-Карпа для пошуку підрядка хешів матриці m на m в масиві хешів матриці n на n.

Для тексту довжини n та шаблону довжини m його середній та найкращий час виконання O(n) при правильному виборі хеш-функци, але в гіршому випадку O(nm)

Тоді для матриць в найгіршому випадку O(n\*n\*m\*m), а в середньому O(n\*n)

1. **Основні модулі програми**

//вивід на екран

* void print(char\*\* matrix)

//порівняння матриць починаючи від заланих індексів

* bool compMatrix(char\*\* text, char\*\* pattern, int row, int col, int M)

//порівняння рядків починаючи від заланих індексів

* bool comp(int\* s1, int\* s2, int del)

//звичайний метод Рабіна-Карпа

* int RabinKarpSearch(char\*\* text, char\*\* pattern, int \*txt, int \*pat, int row)

//розрахунок хешів

* void culcHashes(char \*\*matrix, int\* hashes, int width, int height)

//перерахунок хешів

* void reculcHash(char \*\*matrix, int\* hashes, int width, int height, int row)

//узагальнення для матриць

* void RabinKarpMatrix(char \*\*text, char \*\*pattern)

//демонстрація роботи програми

* void Rabin\_Karp\_demo()

1. **Джерела**

Кормен «Вступ до алгоритмів»