**Київський національний університет імені Тараса Шевченка**

**Факультет комп’ютерних наук та кібернетики**

**Кафедра інформаційних систем**

**Алгоритми та складність**

**Лабораторний проект № 7**

**Узагальнення методу Рабіна-Карпа**

**Звіт**

**Виконала:**

студентка групи К-28

Кушніренко Марія Сергіївна

**Київ-2018**

**Узагальнення методу Рабіна-Карпа**

1. **Умова 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. **Опис і аналіз алгоритму.**

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-функції, але в гіршому випадку O(nm)

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

1. **Реалізація алгоритму.**

Реалізовано на С++

1. **Інтерфейс програми.**

Дані (тести) прописані в файлі срр, а результат виводиться в консоль.

1. **Використані структури даних.**

Вектори вигляду: vector<string> v, vector<int> v

1. **Тестовий приклад.**

model:

f s f

m n x

c j d

matrix:

h h v v n

d f s f d

c m n x i

n c j d k

c n j d j

Find at row = 2, column = 2

1. **Основні модулі програми.**

//вивід матриці на екран

void Cout(vector<string> v)

//порівняння матриць починаючи від заданих індексів

bool compare\_matrix(vector<string> matrix, vector<string> model, int row, int col, int size)

//порівняння рядків починаючи від заданих індексів

bool compare\_rows(vector<int> s1, vector<int> s2, int ind)

//метод Рабіна-Карпа

int RK(vector<string> matrix, vector<string> model, vector<int> hashMatrix, vector<int> hashModel, int row)

//розрахунок хешів

void culculationHashes(vector<string> v, vector<int> &hashes, int width, int height)

//перерахунок хешів

void reculculationHashes(vector<string> v, vector<int> &hashes, int width, int height, int row)

//узагальнення для матриць

void generalizationRK(vector<string> matrix, vector<string> model)

1. **Використані джерела.**

1.Кормен «Алгоритмы построение и анализ» (ст. 1036)

2.https://uk.wikipedia.org/wiki/%D0%90%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC\_%D0%A0%D0%B0%D0%B1%D1%96%D0%BD%D0%B0\_%E2%80%94\_%D0%9A%D0%B0%D1%80%D0%BF%D0%B0