МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МО ЭВМ

ОТЧЕТ

по лабораторной работе №4

по дисциплине «Алгоритмы и Структуры Данных»

Тема: Поиск образца в тексте: алгоритм Рабина-Карпа

Студент гр. 1303 _____ Коренев Д. А.

Преподаватель _____ Иванов Д. В.

Санкт-Петербург

2022

Цель работы.

Научиться работе с хэш-функцией, с её помощью решить алгоритмическую задачу.

Задание.

Поиск образца в тексте. Алгоритм Рабина-Карпа.

Напишите программу, которая ищет все вхождения строки Pattern в строку Text, используя алгоритм Карпа-Рабина.

На вход программе подается подстрока Pattern и текст Text. Необходимо вывести индексы вхождений строки Pattern в строку Text в возрастающем порядке, используя индексацию с нуля.

Примечание: в работе запрещено использовать библиотечные реализации алгоритмов и структур.

Ограничения

$1 \le |Pattern| \le |Text| \le 5 \cdot 105$.

Суммарная длина всех вхождений образца в текста не превосходит 108. Обе строки содержат только буквы латинского алфавита.

Пример.

Вход:

aba

abacaba

Выход:

0 4

Подсказки:

1. Будьте осторожны с операцией взятия подстроки — она может оказаться дорогой по времени и по памяти.

2. Храните степени x ** p в списке - тогда вам не придется вычислять их каждый раз заново.

Выполнение работы.

Для выполнения работы был создан класс MyString с методом subString(sub), находящий индексы всех подстрочек в основной строке. Для этого был использован алгоритм Рабина-Карпа.

Рассмотрю используемый алгоритм. В методе subString(sub) для каждой подстроки высчитывается значение равное произведению значения буквы (a = 0, b = 1 и тд.) на 2 в определенной степени. Степень двойки равна разнице длины строки без единицы и индекса буквы в подстроки, то и есть наша хэш функция. Далее, когда значение для подстроки подсчитано, значение запоминается. Вызывается метод hashingString — для каждого символа строки высчитывается значение хэш функцией, если значение совпадает со значение подстроки, проверяется совпадают ли подстроки и, при совпадении, индекс добавляется в массив ans. Проверка подстроки требуется так как возможны коллизии: любое число можно представить в виде сумм степеней двоек (т.е. коллизий не было бы, если бы алфавит состоял из двух букв и, соответственно, со значениями в хэш функции 0 и 1), но в латинском алфавите букв намного больше чем две. Для того чтобы посчитать значение для следующего символа в строке для оптимизации из значения хэш функции для прошлого символа вычитается его первое слагаемое (т. к. этот символ отсутствует в новой подстроке), далее это значение умножается на 2 и прибавляется значение следующего символа, вновь производится проверка равенства значений с искомой подстрокой. Полученный массив ans возвращается в метод subString, который тоже возвращает этот массив. В функции main выводится в консоль ответ в соответствии с требованиями.

Тестирование программы.

Чтобы удостовериться в правильности работы программы, она была протестирована на следующих случаях:

• В тексте не встречается требуемой подстроки;

• Подстрока находится на конце текста;

• Подстрока находится в начале текста;

• Подстрока находится в середине текста;

• Подстрока является одной буквой;

• Подстрока равна длине строки.

Выводы.

В ходе выполнения лабораторной работы был изучен алгоритм РабинаКарпа, были применены навыки работы с хэш-функцией. Была успешно решена поставленная задача.