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

ОТЧЕТ

по лабораторной работе №5

по дисциплине «Построение и анализ алгоритмов»

Тема: «Алгоритм Ахо-Корасик»

Студент гр. 7304 _____ Дементьев М.Е.

Преподаватель _____ Филатов А.Ю.

Санкт-Петербург

2019

Цель работы

Изучать алгоритм Ахо-Корасик, его реализацию на языке C++ и применение при решении практических задач на поиск вхождений образов в заданную строку.

Задачи

1) Разработать программу, решающую задачу точного поиска набора образцов с помощью алгоритма Ахо-Корасик.

   Вход:

   Первая строка содержит текст (T,$1 \le |T| \le 100000$).

   Вторая - число n ($1 \le n \le 3000$), каждая следующая из n строк содержит шаблон из набора $P=\{p1,\ldots,pn\}$ $1 \le |pi| \le 75$

   Все строки содержат символы из алфавита {A,C,G,T,N}

   Выход:

   Все вхождения образцов из P в T.

   Каждое вхождение образца в текст представить в виде двух чисел - i p

   Где i - позиция в тексте (нумерация начинается с 1), с которой начинается вхождение образца с номером p

   (нумерация образцов начинается с 1).

   Строки выхода должны быть отсортированы по возрастанию, сначала номера позиции, затем номера шаблона.

2) Используя реализацию точного множественного поиска, решить задачу точного поиска для одного образца с *джокером*.

   Вход:

   Текст (T,$1 \le |T| \le 100000$)

   Шаблон (P,$1 \le |P| \le 40$)

   Символ джокера

   Выход:

Строки с номерами позиций вхождений шаблона (каждая строка содержит только один номер).

Номера должны выводиться в порядке возрастания.

Описание алгоритма

Суть алгоритма заключена в использование структуры данных — бора и построения по нему конечного детерминированного автомата. Строится бор последовательным добавлением исходных строк. Изначально есть 1 вершина, корень - пустая строка. Добавление строки происходит так: начиная в корне, двигаемся по дереву, выбирая каждый раз ребро, соответствующее очередной букве строки. Если такого ребра нет, то мы создаем его вместе с вершиной. Так как процесс добавления строки может остановиться во внутренней вершине, то для каждой строки будем дополнительно хранить признак того является она строкой из условия или нет. Далее, строим конечный детерминированный автомат. Состояние автомата — это какая-то вершина бора. Переход из состояний осуществляется по 2 параметрам — текущей вершине node и символу symb. по которому нам надо сдвинуться из этой вершины. Назовем суффиксной ссылкой вершины v указатель на вершину u, такую что строка u — наибольший собственный суффикс строки v, или, если такой вершины нет в боре, то указатель на корень. В частности, ссылка из корня ведет в него же. Если из текущей вершины есть ребро с символом symb, то пройдем по нему, в обратном случаем пройдем по суффиксной ссылке и запустимся рекурсивно от новой вершины. Алгоритм завершится, когда мы дойдем до конца строки.

Результаты работы программы

1) *Входные данные:*

```
CCCA

1

CC
```

Выходные:

```
1 1
2 1
```

2) *Входные данные:*

```
ACT

A$

$
```

Выходные:

```
1
```

Выводы

В ходе лабораторной работы был изучен алгоритм Ахо-Корасик и решена задача точного поиска набора образцов. В результате выполнения лабораторной работы были разработаны программы, которые вычисляют начальные позиции вхождения простых шаблонов в тексте, а также шаблонов, содержащих символ джокера. В процессе выполнения работы была решена проблема построения суффиксальных ссылок и проблема поиска конечных состояний автомата, построенного из бора.