**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. 6381 |  | Токун М.С. |
| Преподаватель |  | Филатов А. Ю. |

Санкт-Петербург

2018

**Цель работы.**

Решить задачу точного поиска набора образцов с помощью алгоритма Ахо-Корасик, а также решить задачу точного поиска для одного образца с джокером, используя реализацию точного множественного поиска.

**Задание.**

1. Разработайте программу, решающую задачу точного поиска набора образцов.

*Входные данные:*

Первая строка содержит текст (T,1≤|T|≤100000).

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

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

*Выходные данные:*

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.

1. Используя реализацию точного множественного поиска, решите задачу точного поиска для одного образца с джокером.

*Входные данные:*

Текст (T,1≤|T|≤100000)

Шаблон (P,1≤|P|≤40)

Символ джокера

*Выходные данные:*

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

Номера должны выводиться в порядке возрастания.

**Описание алгоритма.**

На вход подается текст и шаблоны, которые необходимо найти. На основе введенных шаблонов строится бор.

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.

Продвигаясь по тексту, выполняем переход по бору, либо по суффиксной ссылке. 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 c текущим символом, то пройдем по нему, иначе пройдем по суффиксной ссылке и запустимся рекурсивно от новой вершины.

Если удалось найти шаблон в тексте, запоминаем позицию начала вхождения, номер шаблона и выводим их на экран, завершение алгоритма происходит, когда мы дойдем до конца строки.

**Описание функций.**

1. **int getMove(int node, char symb)** – получение следующего перехода.
2. **void check(int node, int i)** – поиск полного вхождения паттерна.
3. **void find(const string& text)** – поиск вхождений паттернов в исходном тексте.
4. **void addToBor(const string& pattern)** – добавление строки в бор
5. **int getSuffixLink(int node) –** вычисление суффиксной ссылки;

**Тестирование.**

|  |  |  |
| --- | --- | --- |
| Задание | Данные на вход | Результат работы программы |
| Точный поиск набора образцов в строке | AHFFYYAJAHFAUAHF  1  AHF |  |
|  | CCCA  1  CC |  |
| Точный поиск для одного образца с джокером | ASWEEAWSED  W#E  # |  |
|  | ACT  A$  $ |  |

**Вывод.**

В процессе выполнения лабораторной работы был реализован алгоритм Ахо-Корасик, а также построены программы для точного поиска набора образцов в строке и для точного поиска для одного образца с джокером.