

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. 9383

Рыбников Р.А.

Преподаватель

Фирсов М.А.

Санкт-Петербург

2021

Цель работы

Изучение и реализация алгоритма Ахо-Корасик для поиска вхождений нескольких шаблонов или вхождения шаблона с джокером в текст.

Задание 1

Разработайте программу, решающую задачу точного поиска набора образцов.

Вход:

Первая строка содержит текст ($T, 1 \leq |T| \leq 100000$). Вторая – число n ($1 \leq n \leq 3000$), каждая следующая из n строк содержит шаблон из набора $P = \{p_1, \dots, p_n\}$ $1 \leq |p_i| \leq 75$. 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.

Sample Input:

```
NTAG
3
TAGT
TAG
T
```

Sample Output:

```
2 2
2 3
```

Задание 2

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

В шаблоне встречается специальный символ, именуемый джокером (wild card), который «совпадает» с любым символом. По заданному содержащему шаблоны образцу P необходимо найти все вхождения P в текст T .

Например, образец $ab??c?$ с джокером $?$ встречается дважды в тексте $xabvccbababcax$.

Символ джокер не входит в алфавит, символы которого используются в T .

Каждый джокер соответствует одному символу, а не подстроке неопределённой длины. В шаблон входит хотя бы один символ не джокер, т.е. шаблоны вида ??? недопустимы.

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

Вход:

Текст ($T, 1 \leq |T| \leq 1000000$)

Шаблон ($P, 1 \leq |P| \leq 40$)

Символ джокера

Выход:

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

Номера должны выводиться в порядке возрастания.

Sample Input:

ACTANCA

A\$\$\$A\$

\$

Sample Output:

1

Выполнение работы.

В основе алгоритма используется бор. Бор – это дерево, в котором каждая вершина обозначает строку. Инициализируя бор, в нём находится только корень(пустая строка). Добавление строки в бор происходит так: проходим по дереву(бор), выбирая рёбра для обхода таким образом, чтобы ребро соответствовало очередной букве нашей строки; в случае, если ребра такого не нашлось, мы создаём ребро вместе с вершиной. Когда обработан последний символ, помечаем последнюю вершину конечной и переходим к следующему шаблону, продолжая построение, начиная с корня бора.

Помимо бора, используется автомат, построенный на основе бора, который содержит в себе суффиксные ссылки для каждой вершины.

Суффиксная ссылка – это ссылка на узел, соответствующий самому длинному суффиксу. Для коренной вершины суффиксной ссылкой будет петля. Для остальных вершин суффиксная ссылка создаётся по следующему алгоритму: если вершина не корень, то суффиксная ссылка – это вершина, в которую ведёт ребро с данным символом из суффиксной ссылки родительской вершины. Если текущая суффиксная ссылка – конечная вершина, то конечная суффиксная ссылка найдена.

1. Алгоритм Ахо-Корасик (с джокером).

Для данного алгоритма также строится бор, но не для шаблона, а для джокерных подшаблонов из шаблонной подстроки. Выделяется массив индексов, длина которого равна длине рассматриваемой строки, инициализированный нулями. На основе бора строится автомат, и дальше выполняется посимвольное рассмотрение строки.

Если в строке нашёлся какой-либо подшаблон, то ячейка массива по адресу, образованному разностью номера начального символа данного вхождения подшаблона в строке и его смещения относительно начала исходного шаблона, инкрементируется. Если у подшаблона несколько смещений, то данная операция выполняется для каждого из них.

В итоге индексы тех ячеек массива, значение которых будет равно количеству подшаблонов в исходном шаблоне, и будут индексами вхождения заданного шаблона в строку.

Сложность алгоритма

Сложность по памяти алгоритма Ахо-Корасик – $O(|N|)$, где $|N|$ -- суммарная длина искоемых образцов.

Сложность по времени алгоритма Ахо-Корасик – $O((|T| + |N|)\log(s) + k)$, где T – длина текста, s – размер алфавита, k – число вхождений шаблонов в текст.

Сложность по памяти алгоритма Ахо-Корасик с джокером – $O(|T| + |N|)$

Сложность по времени алгоритма Ахо-Корасик с джокером – $O((|T| + |N|)\log(s) + k*p)$, где p – количество сдвигов подшаблонов относительно исходного шаблона.

Описание функций и структур данных.

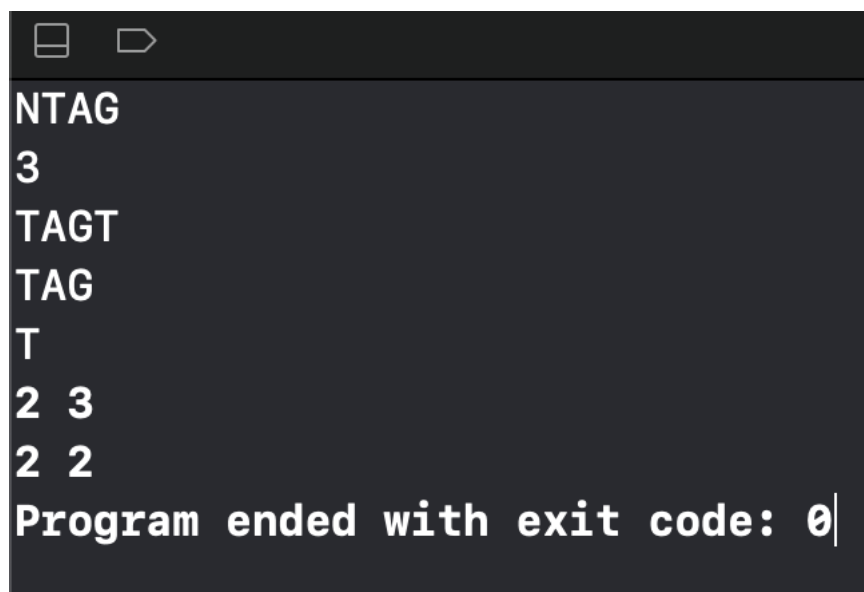
Вершина бора описана структурой `BohrVertex`, имеющей поля, несущие в себе информацию о:

- номере вершины в которой происходит переход по символу
- номер строки в образе
- индекс родителя
- индекс суффиксной ссылки наибольшего суффикса
- индекс символа, по которому производится переход от родителя к текущей вершине
- информация о конце строки

Класс `Bohr` описывает сам бор и автомат по этому же бору, используя методы:

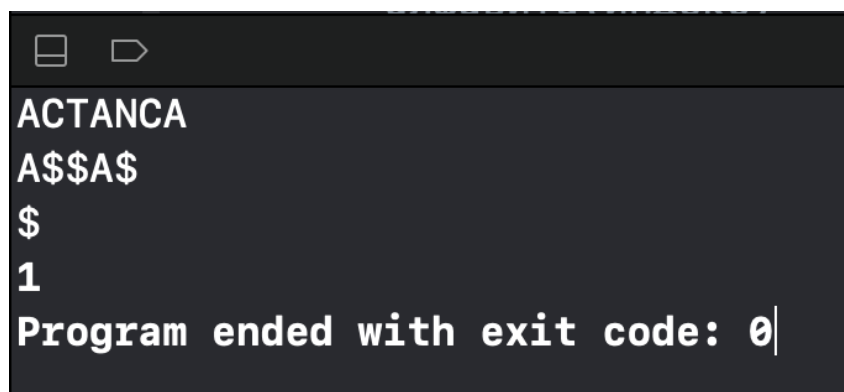
- `void AddInBohr(std::string &str)`
- `int GetSuffix(int vertex)`
- `int Move(int vertex, int symb)` – проход по бору
- `void AhoK(const std::string &str)` – метод для запуска алгоритма Ахо-Корасик

Тестирование



```
NTAG
3
TAGT
TAG
T
2 3
2 2
Program ended with exit code: 0|
```

Рисунок 1 – Тестирование алгоритма без джокера.



```
ACTANCA
A$$$A$
$
1
Program ended with exit code: 0|
```

Рисунок 2 – Тестирование алгоритма с джокером.

Выводы.

Был изучен и реализован алгоритм Ахо-Корасик для поиска вхождений нескольких шаблонов или вхождения шаблона с джокером в текст.