

Московский авиационный институт  
(национальный исследовательский университет)

Факультет информационных технологий и прикладной  
математики

Кафедра вычислительной математики и программирования

Лабораторная работа №5 по курсу «Дискретный анализ»

Студент: Меджидли И. И. о  
Преподаватель: Михайлова С. А  
Группа: М8О-201Б-21  
Дата: 24.02.2024  
Оценка:  
Подпись:

Москва, 2024

## Лабораторная работа №5

**Задача:** Найти в заранее известном тексте поступающие на вход образцы с использованием суффиксного массива.

# 1 Описание

Суффиксный массив строки — это массив индексов ее суффиксов, который задает их порядок в порядке лексикографической сортировки.

Рассмотрим алгоритм построения суффиксного массива:

- 1) Инициализируем вектор, длина которого равна длине строки. Заполняем его значения от нуля до значения длины строки.
- 2) Сортируем полученный вектор с помощью метода `sort` в предикате, который сравнивающего два суффикса - элемента вектора.

Рассмотрим алгоритм поиска образцов в заданном тексте с помощью суффиксного массива:

- 1) Сканируется образец.
- 2) С помощью бинарного поиска в суффиксном массиве берется подстрока исходной строки.
- 3) Если их первые буквы совпадают, то сравнивается весь образец с подстрокой.
- 4) Если они совпали, то порядковый номер суффикса заносится в вектор. Далее по суффиксному массиву берутся смежные подстроки и идет дальнейшие сравнения, до тех пор пока больше не будет совпадений или закончатся элементы суффиксного массива.

Если же образец не совпал с подстрокой, то продолжается бинарный поиск.

- 5) В итоге мы будем иметь вектор индексов начала образца в исходной строке. Все эти шаги будут повторяться, пока поступают образцы.

## 2 Исходный код

```
1 #include <iostream>
2 #include <string>
3 #include <vector>
4 #include <algorithm>
5 #include <numeric>
6
7 using namespace std;
8
9 int main()
10 {
11     ios::sync_with_stdio(false); cin.tie(0);
12     string T, P;
13     cin >> T;
14     int k = 0, ssize = T.length();
15     vector<int> suff(ssize);
16     iota(suff.begin(), suff.end(), 0);
17     sort(suff.begin(), suff.end(),
18         [&T, &ssize](int i, int j) {
19             return T.compare(i, ssize - i, T, j, ssize - j) < 0;
20         });
21     while(cin >> P){
22         ++k;
23         int size = P.length(); int l = 0, r = ssize-1, mid = (r + l) / 2, up = 0, down =
24             0;
25         vector<int> res;
26         while((r - l + 1) > 0){
27             string subik = T.substr(suff[mid]);
28             if(P[0] == subik[0]){
29                 if(P == T.substr(suff[mid], size)) res.push_back(suff[mid]+1);
30
31                 up = mid - 1, down = mid + 1;
32                 if(up != -1){
33                     string subikup = T.substr(suff[up]);
34                     while(P[0] == subikup[0]){
35                         if(P == T.substr(suff[up], size)) res.push_back(suff[up]+1);
36                         --up;
37                         if(up == -1) break;
38                     }
39                 }
40                 if (down != ssize){
41                     string subikdown = T.substr(suff[down]);
42                     while(P[0] == subikdown[0]){
43                         if(P == T.substr(suff[down], size)) res.push_back(suff[down]+1);
44                         ++down;
45                         if(down == ssize) break;
46                     }
47                 }
48             }
49         }
50     }
```

```

47         int ressize = res.size();
48         if(ressize == 0) break;
49         sort(res.begin(), res.end());
50         cout << k << ": ";
51         for(int lk = 0; lk < ressize; ++lk){
52             if(lk == ressize - 1){
53                 cout << res[lk] << "\n";
54                 break;
55             }
56             cout << res[lk] << ", ";
57         }
58         break;
59     }
60
61     else if(P[0] < subik[0]) r = mid - 1;
62
63     else if(P[0] > subik[0]) l = mid + 1;
64
65     mid = (r + l) / 2;
66 }
67 }
68
69 return 0;
70 }

```

### 3 Тесты

Тестировать программу буду ручным способом.

```
imedzhidli@imedzhidli:~/Desktop/DA/LABA5$ ./lab5
abcdabc
abcd
bcd
bc
1: 1
2: 2
3: 2,6
imedzhidli@imedzhidli:~/Desktop/DA/LABA5$ ./lab5
mississippi
si
s
issi
p
1: 4,7
2: 3,4,6,7
3: 2,5
4: 9,10
imedzhidli@imedzhidli:~/Desktop/DA/LABA5$
```

Как можно заметить, все работает верно.

## 4 Выводы

При выполнении пятой лабораторной работы по курсу «Дискретный анализ» я познакомился со способом поиска образцов в тексте, основанном на суффиксных массивах, смог его реализовать. При реализации я использовал бинарный поиск, анонимную функцию и т.д, что позволило мне повторить этот материал.