

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ

ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет «Информатика и вычислительная техника»

Кафедра «Кибербезопасность информационных систем»

Дисциплина «Методы программирования»

Лабораторная работа №2

на тему «Алгоритмы на массивах, сортировка и поиск: скользящее окно,

два указателя, префиксные суммы, стандартная сортировка и классификация алгоритмов сортировки, линейный, бинарный, тернарный поиск»

Выполнила студент группы ВКБ31

Закарян Т.Г.

г. Ростов-на-Дону

2021 г.

Код:

#include <bits/stdc++.h>

using namespace std;

#define in\_range(x,y,z) for (int x=y; x<z; x++)

typedef long long ll;

int find\_subctr(string t, string s)

{

const int p = 97; // основание системы счисления для хеширования строк

// вычисляем степени числа P, чтобы ускорить процесс хеширования

vector<ll> p\_pow (s.length());

p\_pow[0] = 1;

for (size\_t i=1; i<p\_pow.size(); ++i)

p\_pow[i] = p\_pow[i-1] \* p;

// вычисляем хеши всех префиксов строки S

vector<ll> h (s.length());

for (size\_t i=0; i<s.length(); ++i)

{

h[i] = (s[i] - 't' + 1) \* p\_pow[i];

if (i) h[i] += h[i-1];

}

// посчитаем хеш строки Т

ll h\_t = 0;

for (size\_t i=0; i<t.length(); ++i)

h\_t += (t[i] - 't' + 1) \* p\_pow[i];

// сравниваем хеши строки T и подстрок S длины |T|

for (size\_t i = 0; i + t.length() - 1 < s.length(); ++i)

{

ll cur\_h = h[i+t.length()-1];

if (i) cur\_h -= h[i-1];

// если равенство выполняется хотя бы один раз, мы прекращаем поиск

if (cur\_h == h\_t \* p\_pow[i])

return int(i);

}

return -1;

}

int main()

{

vector<string>T,S,C; // наборы строк T и S, а также набор образцов единичной длины C

int n;

cin >> n;

string tmp;

getline(cin,tmp);

in\_range(i,0,n)

{

getline(cin,tmp);

if(tmp.length()==1) C.push\_back(tmp);

T.push\_back(tmp);

}

while(getline(cin,tmp))

{

S.push\_back(tmp);

}

in\_range(i,0,S.size())

{

string s=S[i];

if(s=="") continue; // отсекаем случай пустой строки

if(s.length()==1) // в случае строки единичной длины ищем вхождения образцов из С простым перебором

{

in\_range(j,0,C.size())

{

string c=C[j];

if(s==c)

{

cout << s << "\n";

break;

}

}

continue;

}

in\_range(j,0,n) // во всех остальных случаях ищем вхождения строк из Т, используя алгоритм Рабина-Карпа

{

string t=T[j];

int pos = find\_subctr(t,s);

if(pos>=0)

{

cout << s << "\n";

break;

}

}

}

return 0;

}

Оценка программы:

