МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

Федеральное государственное бюджетное

образовательное учреждение

высшего образования

ВЯТСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ФАКУЛЬТЕТ КОМПЬЮТЕРНЫХ И ФИЗИКО-МАТЕМАТИЧЕСКИХ НАУК

КАФЕДРА ПРИКЛАДНОЙ МАТЕМАТИКИ И ИНФОРМАТИКИ

ОТЧЕТ

ПО ЛАБОРАТОРНОЙ РАБОТЕ №6

**по дисциплине**

**“ Алгоритмы построение и анализ”**

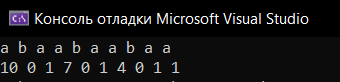
Выполнил: студент гр. ФИб-3302-51-00 ё. \_\_\_\_\_\_\_\_\_\_\_

Проверил: доцент кафедры ПМиИ Разова Е. В. \_\_\_\_\_\_\_\_\_\_\_

Киров 2020

***Задание 1. z-функция***

Реализовать эффективный метод нахождения значений z-функции.



#include<iostream>

#include <string>

using namespace std;

// Заполняет Z-массив для заданной строки str []

void getZarr(string str, int Z[])

{

Z[0] = str.length();

int n = str.length();

int L, R, k;

// [L, R] создать окно, соответствующее префиксу s

L = R = 0;

for (int i = 1; i < n; ++i)

{

// если i> R ничего не подходит, мы рассчитаем. Z [i] наивным способом.

if (i > R)

{

L = R = i;

// R-L = 0 в начале, поэтому проверка будет начинаться с 0-го индекса.

while (R < n && str[R - L] == str[R])

R++;

Z[i] = R - L;

R--;

}

else

{

//k = i-L, поэтому k соответствует числу, которое соответствует интервалу [L, R].

k = i - L;

// если Z [k] меньше оставшегося интервала, то Z [i] будет равно Z [k].

if (Z[k] < R - i + 1)

Z[i] = Z[k];

else

{

// иначе начинаем с R и проверяем вручную

L = i;

while (R < n && str[R - L] == str[R])

R++;

Z[i] = R - L;

R--;

}

}

}

}

int main()

{

string text;

text = "abaabaabaa";

//cin >> text;

int Z[100];

getZarr(text, Z);

for (int i = 0; i < text.length(); i++)

cout << text[i] << " ";

cout << endl;

for (int i = 0; i < text.length(); i++)

cout << Z[i] << " ";

return 0;

}

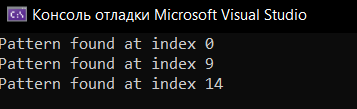
***Задание 2. Алгоритм поиска подстроки в строке на основе использования z-функции***

Дана цепочка S и образец P. Требуется, используя z-функцию,

А) найти первое вхождение P в S;

Б) найти последнее вхождение P в S;

С) найти все позиции, начиная с которых P входит в S.



#include<iostream>

#include <string>

using namespace std;

void getZarr(string str, int Z[]);

// печатает все вхождения шаблона в текст, используя алгоритм Z

void search(string text, string pattern)

{

string concat = pattern + "$" + text;

int l = concat.length();

// Построить Z-массив

int Z[1000];

getZarr(concat, Z);

// теперь перебираем массив Z для соответствия условию

for (int i = 0; i < l; ++i)

{

// если Z [i] (совпадающая область) равна длине подстроки, то это подстрока

if (Z[i] == pattern.length())

cout << "Pattern found at index "

<< i - pattern.length() - 1 << endl;

}

}

void getZarr(string str, int Z[])

{

int n = str.length();

int L, R, k;

L = R = 0;

for (int i = 1; i < n; ++i)

{

if (i > R)

{

L = R = i;

while (R < n && str[R - L] == str[R])

R++;

Z[i] = R - L;

R--;

}

else

{

k = i - L;

if (Z[k] < R - i + 1)

Z[i] = Z[k];

else

{

L = i;

while (R < n && str[R - L] == str[R])

R++;

Z[i] = R - L;

R--;

}

}

}

}

int main()

{

string text = "AABACABABAABACAABA";

string pattern = "AABA";

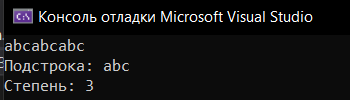
search(text, pattern);

return 0;

}

***Задание 3.  Сжатие строки***

Решить задачу сжатия строки, используя z-функцию. Описание задачи см. в лабораторной работе №4.



#include<iostream>

#include <string>

#include <clocale>

#include <vector>

using namespace std;

int countnews = 0;

vector<int> ZarrFindSubstring(const string& s)

{

//cout << s;

int len = s.length();

vector<int> p(len);

int k = 0; // счетчик

for (int i = 1; i < len; i++)

{

if (s[k] != s[i])

{

// повторная проверка при k = 0

if (s[0] != s[i]) k = 0;

else k = 1;

}

else

{

k++; // если символы совпадают -> увеличиваем значение счетчика

}

p[i] = k; // значение счетчик в вектор

}

for (int i = 0; i < len; i++) {

//cout << " " << p[i] << " ";

if (p[i] == 0) countnews++;

}

//cout << endl;

return p;

}

void getZarr(string str, int Z[])

{

Z[0] = str.length();

int n = str.length();

int L, R, k;

L = R = 0;

for (int i = 1; i < n; ++i)

{

cout << endl;

if (i > R)

{

L = R = i;

while (R < n && str[R - L] == str[R])

R++;

Z[i] = R - L;

R--;

}

else

{

k = i - L;

if (Z[k] < R - i + 1)

Z[i] = Z[k];

else

{

L = i;

while (R < n && str[R - L] == str[R])

R++;

Z[i] = R - L;

R--;

}

}

}

string tmps = "";

for (int i = 0; i < str.length(); i++)

cout << str[i] << " ";

cout << endl;

for (int i = 0; i < str.length(); i++)

cout << Z[i] << " ";

cout << endl;

cout << endl;

int len = str.length();

int answer = 0;

for (int i = 0; i < len; i++) {

tmps += str[i];

reverse(tmps.begin(), tmps.end());

ZarrFindSubstring(tmps); //вовзращает кол-во новых

reverse(tmps.begin(), tmps.end());

answer += countnews;

countnews = 0;

}

cout << endl << "Количество непустых подстрок: "<<answer << endl;

}

int main()

{

setlocale(0, "");

string text;

cin >> text;

int Z[100];

getZarr(text, Z);

return 0;

}