Федеральное государственное автономное

образовательное учреждение

высшего профессионального образования

«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

|  |
| --- |
| Институт космических и информационных технологий |
| институт |
| Кафедра «Информатики» |
| кафедра |

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ**

|  |
| --- |
| Поиск образа в строке по алгоритму Боуэра-Мура |
| Тема / Аббревиатура, слово-символ (слово-бренд) |
|  |
|  |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Преподаватель |  |  | | Д.В. Грузенкин | | | | | / | | |  | |
|  | дата | |  | | инициалы, фамилия | | | | |  | | | подпись | |
| Студент | КИ14-16б | | | | |  |  |  | | | С.В. Кухаренко | | | | |  |  |
|  | код (номер) группы | | | | |  | дата |  | | | инициалы, фамилия | | | | |  | подпись |

Красноярск 2015

**Постановка задачи**

Программно реализовать алгоритмы поиска образа в строке: алгоритм Кнута-Морриса-Пратта (КМП), алгоритм Боуэра-Мура (БМ). Программу первого алгоритма разработал Скрипкин Михаил, второго – Кухаренко Сергей.

**Описание алгоритма**

Алгоритм Боуэра-Мура:

Данный алгоритм имеет похожую идею: как и в КМП образ смещается вдоль строки слева направо, однако проверка совпадения образа с текущей частью строки происходит справа налево.

Для определения смещения перед поиском происходит создание матрицы d, которая состоит из 256 ячеек и создается с помощью таблицы кодов ASCII.

На начальном этапе все элементы данного целочисленного одномерного массива заполняются величиной длины искомого образа. Затем каждому элементу таблицы d, с индексом, равным коду ASCII рассматриваемого символа образа, присваивается значение удаленности этого символа от конца образа. При этом, если образ содержит несколько одинаковых символов, то элементу таблицы d, соответствующему данному символу, должно быть присвоено значение удаленности от конца образа до самого правого из них.

На втором этапе алгоритма происходит непосредственно поиск образа в строке. Сравнение продолжается до тех пор, пока искомый образ не найден, либо пока не обнаружено несовпадение, либо пока не достигнут конец строки. Если обнаружено несовпадение символа строки и символа образа тогда происходит смещение на величину, равную значению ячейки таблицы d, индекс которой равен коду ASCII символа строки.

**Тексты программ**

Для реализации алгоритма БМ использовался язык программирования C++, среда разработки – Microsoft Visual Studio 2013 Express.

#include "stdafx.h"

#include <Windows.h>

#include <string>

#include <iostream>

using namespace std;

int \_tmain(int argc, \_TCHAR\* argv[])

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

cout << "Пожалуйста, введите строку: ";

string stroka;

getline(cin, stroka, '\n');

cout << "Пожалуйста, введите искомый образ: ";

string obraz;

getline(cin, obraz, '\n');

int d[256]; //Матрица смещения

int len = obraz.length();

int lenStroka = stroka.length();

for (int i = 0; i < 256; i++) //Заполняем матрицу смещения длиной образа

d[i] = len;

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

{

char currSym = obraz[len - 1 - i]; //Идем по образу справа на лево

if (d[(int)currSym] == len) //Если в матрице смещения на позиции кода текущего символа образа записана длина образа (этот символ еще не встречался)

{

d[(int)currSym] = len - (len - i); //тогда записываем на его место расстояние от текущего символа образа до конца образа

}

}

int countRepeat = 1;

for (int i = 1; i < len; i++) //Алгоритм по проверки образа на однородность

{

if (obraz[i] == obraz[i - 1])

countRepeat++;

if (countRepeat == len) //Если строка состоит только из одинаковых символов,

for (int j = 0; j < 256; j++) //Заполняем матрицу смещения единицами, это не даст возможность образу проскочить позицию в строке

d[j] = 1;

int k, sm = 1, j = -1, n = 0;

do

{

j += sm; //Определяет смещение образа по строке

k = 1; //Определят количество совпавших элементов

while ((k <= len) && (obraz[len-k] == stroka[len-k+j]))

k++;

if (k - 1 != len)

{

sm = d[(int)stroka[len - k + j]]; //Смещения согласно матрицы смещения

n++; //Увеличение счетчика смещений (для вычисления эффективности алгоритма)

}

else

{

cout << "Позиция первого вхождения: " << j + 1 << endl;

cout << "Количество смещений: " << n << endl;

}

} while ((k-1 != len) && (j < lenStroka - len));

if (k-1 != len)

printf("Образ не найден\n"); //Строка не найдена

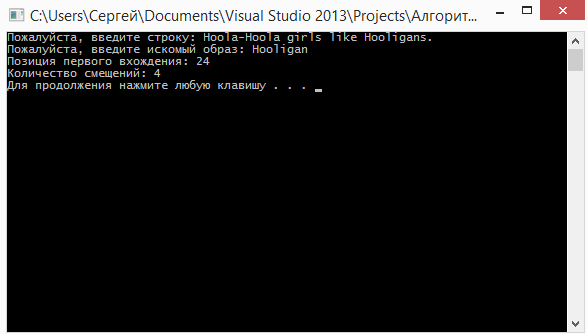
system("pause");

return 0;

}

**Результаты работы программ**

Программа сообщает пользователю позицию первого вхождения образа в строку, а также выводит количество смещений, за которое данный образ был найден.



**Оценки эффективности алгоритмов на разных примерах**

Пример 1:

Строка: «Hoola-Hoola girls like Hooligans»

Образ: «Hooligan»

Алгоритм КМП – 17 смещений.

Алгоритм БМ – 4 смещения.

Пример 2:

Строка: «Mr. Simpson ate my son»

Образ: «son»

Алгоритм КМП – 8 смещений.

Алгоритм БМ – 3 смещения.

Пример 3:

Строка: «abcabq abcabw abcabd abcabg abcabj abcabd»

Образ: «abcabd»

Алгоритм КМП – 4 смещения

Алгоритм БМ – 4 смещения.