МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**

**(ФГБОУ ВО «КубГУ»)**

**Факультет компьютерных технологий и прикладной математики**

**Кафедра информационных технологий**

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 6**

**по дисциплине  
 «АЛГОРИТМЫ И АНАЛИЗ СЛОЖНОСТИ»**

Выполнил студент группы 24/1                                       А.А.Иванов

Направление подготовки  02.03.03  Математическое обеспечение и администрирование информационных систем

Курс    2

Отчет принял доктор физико-математических наук, профессор                                                                                       А.И. Миков

Краснодар

2022 г.

**Задание:** найти зависимость ошибки алгоритма Флажоле-Мартина от величины множества URL (|URL| = m), а так же от длительности наблюдения за потоком |MU| = s. Значение m: 10, 20, 30, 40, 50, 100, 500, 1000. Значение s: 2m, 5m, 10m, 100m.

**Решение:** напишем функцию RandomURL для создания случайных URL-адресов, Hash для нахождения хэш-кода нашего URL, Binary для перевода хэш-кода в бинарный вид, Rang для нахождения ранга хэш-кода. Функции PustMass, FindR и Finder нужны для корректной работы алгоритма Флажоле-Мартина.

**Текст программы:**

#define \_CRT\_SECURE\_NO\_WARNINGS

#include<iostream>

#include<string>

#include<random>

#include<ctime>

#include<cstdlib>

#include<algorithm>

#include<math.h>

//m = 10, 20, 30, 40, 50, 100, 500, 1000

//s = 2m, 5m, 10m, 100m

using namespace std;

const double A = 0.618033;

const double FI = 0.77351;

int n;

struct Table

{

string\* arrs = new string[n];

int\* arrh = new int[n];

string\* arrb = new string[n];

int\* arrr = new int[n];

};

string RandomURL(string &s)

{

s += "https://";

char Symbols[26] = { 'a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's', 't', 'u', 'v', 'w', 'x', 'y', 'z' };

int Length = rand() % 20 + 10;

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

{

int Symbol = rand() % 26;

s += Symbols[Symbol];

}

string s1 = ".com", s2 = ".ru", s3 = ".net";

int Domen = rand() % 3;

if (Domen == 0)

s += s1;

if (Domen == 1)

s += s2;

if (Domen == 2)

s += s3;

return s;

}

int Hash(string &s)

{

int N = 2000, h = 0, k = 0;

while (s[k] != '\0')

{

h += s[k];

k++;

}

h %= N;

return h;

}

string Binary(unsigned s)

{

string s1;

do

{

s1.push\_back('0' + (s & 1));

}

while (s >>= 1);

reverse(s1.begin(), s1.end());

return s1;

}

int Rang(int h)

{

if (h % 1024 == 0)

return 10;

if (h % 512 == 0)

return 9;

if (h % 256 == 0)

return 8;

if (h % 128 == 0)

return 7;

if (h % 64 == 0)

return 6;

if (h % 32 == 0)

return 5;

if (h % 16 == 0)

return 4;

if (h % 8 == 0)

return 3;

if (h % 4 == 0)

return 2;

if (h % 2 == 0)

return 1;

if (h % 2 != 0)

return 0;

}

void PustMass(int\* arr)

{

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

arr[i] = 0;

}

int FindR(int\* arr)

{

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

if (arr[i] == 0)

return i;

}

void Finder(int\* arr, int n1, int s, int N)

{

int R = FindR(arr);

cout << "Для S" << n1 << " = " << s << " \* " << N << ": \n" << endl;

double M = pow(2, R) / FI;

cout << "R = " << R << endl;

cout << "M = " << M << endl;

double E = N - M;

cout << "E = " << E << endl;

double D = E / N;

cout << "D = " << D << "\n" << "\n" << endl;

}

int main()

{

setlocale(LC\_ALL, "rus");

srand(time(NULL));

n = 30;

Table HashTable;

int N = 10;

int max = pow(2, n);

int\* arr = new int[n];

while (N <= 1000)

{

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

{

string s;

HashTable.arrs[i] = RandomURL(s);

HashTable.arrh[i] = Hash(s);

HashTable.arrb[i] = Binary(HashTable.arrh[i]);

HashTable.arrr[i] = Rang(HashTable.arrh[i]);

int s1 = 2 \* N;

PustMass(arr);

for (int j = 0; j < s1; j++)

{

int Temp = rand() % N;

int hash = Hash(s);

int Count = 0;

Rang(hash);

arr[Count] = 1;

}

Finder(arr, 2, 5, N);

int s3 = 10 \* N;

PustMass(arr);

for (int j = 0; j < s3; j++)

{

int Temp = rand() % N;

int hash = Hash(s);

int Count = 0;

Rang(hash);

arr[Count] = 1;

}

Finder(arr, 3, 10, N);

int s4 = 100 \* N;

PustMass(arr);

for (int j = 0; j < s4; j++)

{

int Temp = rand() % N;

int hash = Hash(s);

int Count = 0;

Rang(hash);

arr[Count] = 1;

}

Finder(arr, 4, 100, N);

if (N < 50)

N += 10;

else

if (N < 100)

N += 50;

else

if (N < 500)

N += 400;

else

N += 500;

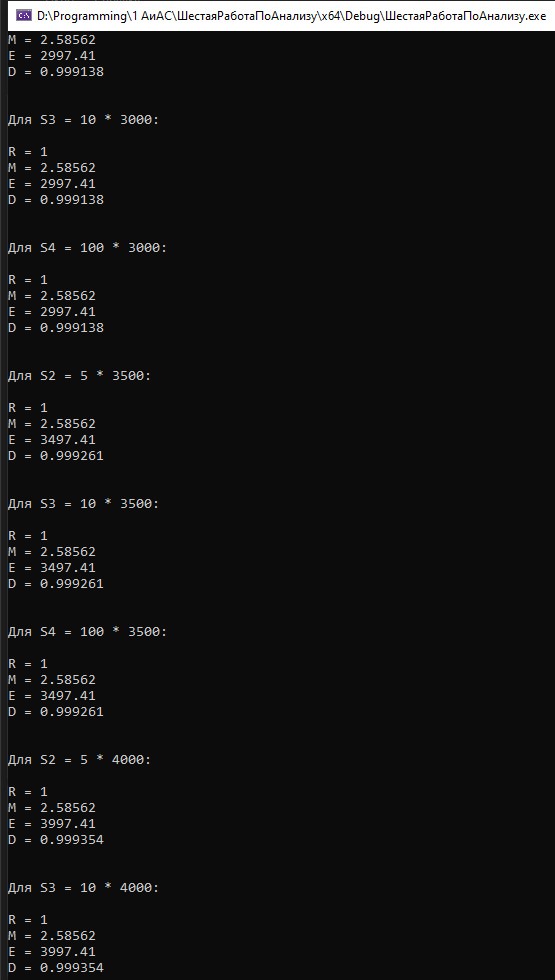
}

}

system("pause");

}

**Пример вывода в консоль:**



**Графики** (используют средние значения):

Для изменения количества Е (m, s) при s = 2m.

|  |  |
| --- | --- |
| 10 | 2997 |
| 20 | 3197 |
| 30 | 3297 |
| 40 | 3997 |
| 50 | 3897 |
| 100 | 4047 |
| 500 | 4347 |
| 1000 | 4647 |

Для изменения количества D (m, s) при s = 2m.

|  |  |
| --- | --- |
| 10 | 0,99 |
| 20 | 0,99 |
| 30 | 0,98 |
| 40 | 0,97 |
| 50 | 0,96 |
| 100 | 0,98 |
| 500 | 0,95 |
| 1000 | 0,98 |

**Вывод**: пространственная сложность алгоритма Флажоле-Мартина = O(log |U|).

Однако результат работы программы может немного отличаться от запуска к запуску (по этому в графиках используются средние значения).