ё

### Федеральное государственное автономное образовательное учреждение высшего образования

# «Национальный исследовательский университет ИТМО»

Факультет среднего профессионального образования

Санкт-Петербург 2019

Специальность: 09.02.07 Информационные системы и программирование

ОТЧЕТ

ПО ЛАБОРАТОРНОЙ РАБОТЕ №5

по теме: Алгоритмы поиска

Проверил:

\_\_\_\_\_\_\_\_ Антонов М.Б.

Дата: “\_\_” \_\_\_\_\_\_ 2019г.

Оценка:\_\_\_\_\_\_\_\_\_\_\_

**Цель работы:** реализовать и замерить скорость работы алгоритмов поиска на больших структурах данных.

Выполнил:

Студент группы Y2333

\_\_\_\_\_\_\_\_\_\_\_\_\_Максимов Д.О

**Задачи:**

* изучить несколько алгоритмов поиска элементов в массиве;
* научиться оценивать сложность программы;
* научиться устанавливать метрики на программное обеспечение.

**Код программы:**

Файл main.cpp:

#include <iostream>

#include <fstream>

#include <cstdlib>

#include <ctime>

using namespace std;

int\* zapolnaem(int m)

{

srand(time(NULL)\*m);

int \*M=new int[m];

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

M[i]= rand() % 6000000;

}

return M;

}

int trans(int\* A, int c, int a){

int i=0;

while ((A[i]!=c)&&(i<a)) {i++; }

if ((i==a)&&(A[i]!=c))

{

return (-1);

}

else {

if (i!=0){

int t;

t = A[i]; A[i] = A[i-1]; A[i-1] = t;

return i;

}

}

}

int binarysearch(int\* A, int a, int n)

{

int low, high, middle;

low = 0;

high = n - 1;

while (low <= high)

{

middle = (low + high) / 2;

if (a < A[middle])

high = middle - 1;

else if (a > A[middle])

low = middle + 1;

else

return middle;

}

return -1;

}

int \*sorte (int\* x, int n)

{

int i,j,r,t;

for (r = n/2; r>0; r/= 2)

{

for (i = r; i < n; i++)

{

t = x[i];

for (j = i; j >= r; j=j-r)

{

if (t < x[j - r])

x[j] = x[j - r];

else

break;}

x[j] = t;}}

return x;

}

int main()

{

srand(time(NULL));

int \*arr = new int[1000000], keys[40], answers[40];

long int start, finish;

for(int i = 0; i < 1000000; i++)

arr[i]=rand()\*(6000000/RAND\_MAX) + rand()%(6000000/RAND\_MAX);

for(int i = 0; i < 40; i++)

keys[i]=rand()\*(6000000/RAND\_MAX) + rand()%(6000000/RAND\_MAX);

start=clock();

for(int i = 0; i<40; i++)

answers[i]=trans(arr, 1000000, keys[i]);

finish=clock();

cout << "Indexes: ";

for(int i = 0; i<40; i++)

cout << answers[i] << " ";

cout << endl << "Linear search with sentinel: " << finish-start << endl;

start=clock();

sorte(arr, \*(arr+1000000));

finish=clock();

cout << endl << "Sorting time: " << finish-start << endl << endl;

start=clock();

for(int i = 0; i<40; i++)

answers[i]=binarysearch(arr, 1000000, keys[i]);

finish=clock();

cout << "Indexes: ";

for(int i = 0; i<40; i++)

cout << answers[i] << " ";

cout << endl << "Binary search: " << finish-start << endl;

return 0;

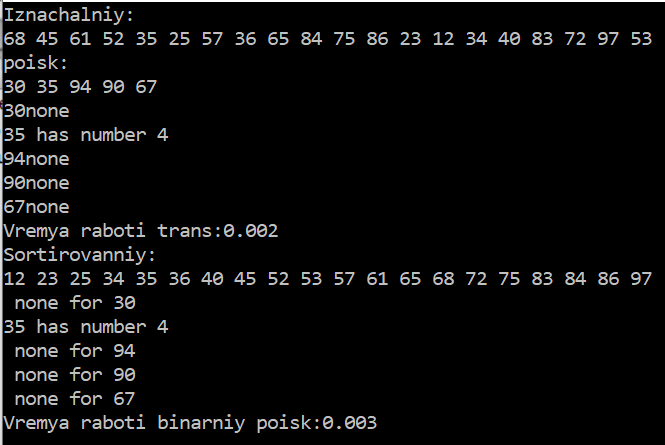
}

**Протокол программы:**

На рисунке 1 представлен скриншот выходного файла. Для проверки был взят массив из 20 элементов, где происходит поиск 5 элементов, представлен на рисунке 2.



*Рисунок 1 – Результат работы программы*

**

*Рисунок 2 – Выходной файл проверки*

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Алгоритм | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |  |
| Перебор с транспозицией | 0.002с | 0.004с | 0.003с | 0.005с | 0.005с | 0.004с | 0.003с | 0.005с | 0.004с | 0.003с |
| Метод бинарного деления | 0с | 0.001с | 0с | 0.001с | 0.001с | 0с | 0.001с | 0.001с | 0.001с | 0с |

Сложность бинарного поиска: O(log n).

Сложность перебора с транспозицией: O(n).

**Вывод:** в ходе выполнения работы были реализованы и замерена скорость работы алгоритмов поиска. Изучены разные алгоритмы поиска.