09.02.07 Информационные системы и программирование

Проверил:

\_\_\_\_\_\_\_\_\_\_\_ Антонов М.Б.

Дата:”\_\_\_” \_\_\_\_\_\_\_20\_\_г.

Оценка: \_\_\_\_\_

Выполнил:

студент группы Y2331

\_\_\_\_\_\_\_\_\_\_\_ Ильченко Р.М.

Лабораторная работа №7

Тема: «Алгоритмы поиска»  
по дисциплине: «Разработка программных модулей»

Санкт-Петербург 2019

УНИВЕРСИТЕТ ИТМО

Факультет среднего профессионального образования

1. цель

Понять работу алгоритмов поиска.

1. задача

Выполнить задачу в соответствии с вариантом.

1. задание

Реализовать алгоритм поиска при помощи массива, предусмотреть возможные ошибки

1. Код

#include <iostream>

#include <cstdlib>

#include <ctime>

#include <math.h>

#include "sort.h"

#include "search.h"

#define SIZE1 40

#define SIZE2 1000000

using namespace std;

int main()

{

int\* A = new int[SIZE1];

int\* A1 = new int[SIZE2];

int key, keyI[SIZE1];

for (int i = 0; i < SIZE1; i++)

A[i] = rand() % 60000;

for (int i = 0; i < SIZE2; i++)

A1[i] = rand() % 60000;

shellSort(A1, SIZE2);

float start = clock();

cout << "GO!!" << endl;

for (int i = 0; i < SIZE1; i++) {

keyI[i] = binSearch(A1, A[i], SIZE2 );

//keyI[i] = simpleSearch(A1, A[i], SIZE2);

}

float finish = clock();

for (int i = 0; i < SIZE1; i++) {

cout << keyI[i] << ", ";

}

cout << "TIME: " << (finish - start);

return 0;

}

Файл сортировки

#include <iostream>

#include <cstdlib>

#include <ctime>

#include <math.h>

using namespace std;

void insertSort(int\* Arr, int size) {

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

for (int j = i; j > 0 && Arr[j] < Arr[j - 1]; j--) {

swap(Arr[j], Arr[j - 1]);

}

}

}

void shellSort(int\* Arr, int SIZE) {

int number = floor(SIZE / 2);

while (number >= 1) {

for (int j = 0; j < number; j++) {

int l = 0;

int\* A = new int[floor(SIZE / number)];

for (int k = j; k < SIZE; k += number) {

A[l] = Arr[k];

l++;

}

insertSort(A, floor(SIZE / number));

l = 0;

for (int k = j; k < SIZE; k += number) {

Arr[k] = A[l];

l++;

}

}

number = number / 2;

}

}

Файл поиска

#include <iostream>

#include <cstdlib>

#include <ctime>

#include <math.h>

using namespace std;

int binSearch(int\* Arr, int key, int SIZE) {

int r = SIZE;

int l = -1;

int m = floor(SIZE / 2);

while (r - l > 1) {

m = floor((l + r) / 2);

if (Arr[m] > key) {

r = m;

}

else {

l = m;

}

}

if (Arr[m - 1] == key)

return m - 1;

else return -1;

}

int simpleSearch(int\* A, int key, int SIZE) {

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

if (A[i] == key) {

swap(A[i], A[1]);

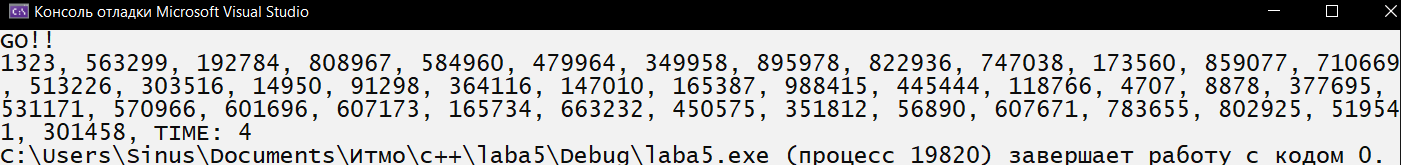
return i;

}

}

return -1;

1. Листинг



1. Вывод

В ходе выполнения лабораторной №5 были получены навыки работы с алгоритмами поиска.