Министерство образования и науки Российской Федерации

Севастопольский государственный университет

Кафедра ИС

Отчет

по лабораторной работе № 6

«Анализ алгоритмов сортировки»

Выполнил: ст. гр. И12д

Серегин А.В.

Проверил:

ст.пр. Шишкевич В.Е.

Севастополь

2015

1. Цель работы

Научиться оценивать сложность и количество операций для алгоритмов сортировки.

2. Варианты заданий

Вариант 6: метод пузырька, сортировка вставками, сортировка Шелла. Отсортировать структуру Время по убыванию.

3. Код программы

main.cpp

#include <iostream>

#include <conio.h>

#include <Windows.h>

#include "Time.h"

using namespace std;

void printTime(const Time time)

{

cout << time.getHour() << "h. "

<< time.getMinute() << "m. "

<< time.getSecond() << "s." << endl;

}

void randTimeArr(Time\*\* arr, const unsigned int n)

{

delete[] \*arr;

\*arr = new Time[n];

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

(\*arr)[i].random();

}

void sortTimeArr(Time\*\* arr, const unsigned int n)

{

for (int step(n / 2); step > 0; step /= 2)

for (int i(0); i < (n - step); i++)

for (int j(i); (j >= 0) && ((\*arr)[j] < (\*arr)[j + step]); j--)

swap((\*arr)[j], (\*arr)[j + step]);

}

int main()

{

setlocale(LC\_ALL, "Russian");

Time\* arr = nullptr;

unsigned int n=0;

char key;

do

{

system("cls");

cout << "1. RandTimeArray" << endl;

cout << "2. PrintTimeArray" << endl;

cout << "3. SortTimeArray" << endl;

cout << "Esc. Exit" << endl;

key = \_getch();

system("cls");

switch (key)

{

case '1':

cout << "Input n:" << endl;

cin >> n;

randTimeArr(&arr, n);

break;

case '2':

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

printTime(arr[i]);

system("pause");

break;

case '3':

cout << "Array is sorted..." << endl;

ULONGLONG stTime = GetTickCount64();

sortTimeArr(&arr, n);

ULONGLONG endTime = GetTickCount64();

cout << "Done. " << endTime-stTime <<"ms." << endl;

system("pause");

break;

}

} while (key != 27);

delete[] arr;

return 0;

}

Time.h

#pragma once

class Time

{

private:

unsigned short int hour\_;

unsigned short int minute\_;

unsigned short int second\_;

public:

Time();

Time(const int);

Time(const unsigned short int, const unsigned short int,

const unsigned short int);

unsigned short int getHour() const;

unsigned short int getMinute() const;

unsigned short int getSecond() const;

unsigned int getTimeInSec() const;

void setHour(const unsigned short int);

void setMinute(const unsigned short int);

void setSecond(const unsigned short int);

void random();

friend bool operator<(const Time, const Time);

friend bool operator>(const Time, const Time);

friend bool operator==(const Time, const Time);

};

Time.cpp

#include "Time.h"

#include <cstdlib>

Time::Time()

{

}

Time::Time(const int second)

{

second\_ = second % 60;

minute\_ = second / 60 % 24;

hour\_ = second / 3600;

}

Time::Time(const unsigned short int hour,

const unsigned short int minute,

const unsigned short int second)

{

second\_ = second;

minute\_ = minute;

hour\_ = hour;

}

unsigned short int Time::getHour() const

{

return hour\_;

}

unsigned short int Time::getMinute() const

{

return minute\_;

}

unsigned short int Time::getSecond() const

{

return second\_;

}

unsigned int Time::getTimeInSec() const

{

return hour\_ \* 3600 + minute\_ \* 60 + second\_;

}

void Time::setHour(unsigned short int hour)

{

hour\_ = hour;

}

void Time::setMinute(unsigned short int minute)

{

minute\_ = minute;

}

void Time::setSecond(unsigned short int second)

{

second\_ = second;

}

void Time::random()

{

hour\_ = rand() % 24;

minute\_ = rand() % 60;

second\_ = rand() % 60;

}

bool operator<(const Time t1, const Time t2)

{

return t1.getTimeInSec() < t2.getTimeInSec();

}

bool operator>(const Time t1, const Time t2)

{

return t1.getTimeInSec() > t2.getTimeInSec();

}

bool operator==(const Time t1, const Time t2)

{

return t1.getTimeInSec() == t2.getTimeInSec();

}

4. Результаты выполнения

На рисунке ниже представлен тестовый пример работы программы, показывающий время выполнения сортировки методом Шелла массива структур Time состоящем из 10000 элементов.

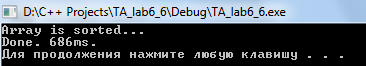


Рисунок 1 – время выполнения сортировки.

Выводы

В ходе лабораторной работы мы узнали что из 3х представленных в задании сортировок самой быстрой является сортировка методом Шелла. Она представлена в моей программе функцией void sortTimeArr(Time\*\*,const int).