Министерство образования Российской Федерации

Пензенский государственный университет

Кафедра «Вычислительная техника»

**ОТЧЕТ**

по лабораторной работе №2

по курсу «Логика и основы алгоритмизации в инженерных задачах»

на тему «Оценка времени выполнения программ»

Выполнили:

студенты группы 21ВВ1.2

Сагателов А.К  
Митрошин А.Д

Приняли:

Митрохин М. А.

Юрова О. В.

Пенза 2022

**Название**

Оценка времени выполнения программ.

**Лабораторное задание.**

Задание 1:

1. Вычислить порядок сложности программы (О-символику).
2. Оценить время выполнения программы и кода, выполняющего перемножение матриц, используя функции библиотеки time.h для матриц размерами от 100, 200, 400, 1000, 2000, 4000, 10000.
3. Построить график зависимости времени выполнения программы от размера матриц и сравнить полученный результат с теоретической оценкой.

Задание 2:

1. Оценить время работы каждого из реализованных алгоритмов на случайном наборе значений массива.
2. Оценить время работы каждого из реализованных алгоритмов на массиве, представляющем собой возрастающую последовательность чисел.
3. Оценить время работы каждого из реализованных алгоритмов на массиве, представляющем собой убывающую последовательность чисел.
4. Оценить время работы каждого из реализованных алгоритмов на массиве, одна половина которого представляет собой возрастающую последовательность чисел, а вторая, – убывающую.
5. Оценить время работы стандартной функции qsort, реализующей алгоритм быстрой сортировки на выше указанных наборах данных.

**Листинг**

#include<iostream>

usingnamespacestd;

voidosnov(int n) {

double time1, time2;

time1 = clock();

int matr1 = newint\* [n];

int matr2 = newint\* [n];

int\*\* C = newint\* [n];

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

{

matr1[i] = newint[n]; // Создаемэлементы

matr2[i] = newint[n]; // Создаемэлементы

C[i] = newint[n]; // Создаемэлементы

}

// А дальше работа как с обычным массивом.

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

{

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

{

matr1[i][j] = rand() % 101 - 50; // Каждый элемент случайному числу от 0 до 9

matr2[i][j] = rand() % 101 - 50;

}

}

for (inti = 0; i< n; i++) {

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

C[i][j] = 0;

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

C[i][j] += matr1[i][k] \* matr2[k][j];

}

}

time2 = clock();

cout<<endl<<"Времяоперациивсекундах: "<< (time2 - time1) / CLOCKS\_PER\_SEC;

}

intmain() {

srand(time(NULL));

setlocale(0, "");

cout<<endl<<"Кол-воэлементов: "<< 100;

osnov(100);

cout<<endl<<"Кол-во элементов: "<< 200;

osnov(200);

cout<<endl<<"Кол-во элементов: "<< 400;

osnov(400);

cout<<endl<<"Кол-во элементов: "<< 1000;

osnov(1000);

cout<<endl<<"Кол-во элементов: "<< 2000;

osnov(2000);

cout<<endl<<"Кол-во элементов: "<< 4000;

osnov(4000);

cout<<endl<<"Кол-во элементов: "<< 10000;

osnov(10000);

system("pause");

}

#include"stdafx.h"

#include<iostream>

#include<cstdlib>

#include<iomanip>

usingnamespacestd;

voidshell(int\* items, int count)

{

inti, j, gap, k;

int x, a[5];

a[0] = 9; a[1] = 5; a[2] = 3; a[3] = 2; a[4] = 1;

for (k = 0; k < 5; k++) {

gap = a[k];

for (i = gap; i< count; ++i) {

x = items[i];

for (j = i - gap; (x < items[j]) && (j >= 0); j = j - gap)

items[j + gap] = items[j];

items[j + gap] = x;

}

}

}

voidqs(int\* items, longint left, longint right) //вызовфункции: qs(items, 0, count-1);

{

longinti, j;

longint x, y;

i = left; j = right;

/\* выбор компаранда \*/

x = items[(left + right) / 2];

do {

while ((items[i] < x) && (i< right)) i++;

while ((x < items[j]) && (j > left)) j--;

if (i<= j) {

y = items[i];

items[i] = items[j];

items[j] = y;

i++; j--;

}

} while (i<= j);

if (left < j) qs(items, left, j);

if (i< right) qs(items, i, right);

}

intcompare(constvoid\* x1, constvoid\* x2)

{

return (\*(int\*)x1 - \*(int\*)x2);

}

voidosnova(int n) {

int\* arr11 = newint[n];

int\* arr12 = newint[n];

int\* arr13 = newint[n];

int\* arr21 = newint[n];

int\* arr22 = newint[n];

int\* arr23 = newint[n];

int\* arr31 = newint[n];

int\* arr32 = newint[n];

int\* arr33 = newint[n];

int\* arr41 = newint[n];

int\* arr42 = newint[n];

int\* arr43 = newint[n];

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

{

arr11[i] = 10 + rand() % 90;

arr12[i] = 10 + rand() % 90;

arr13[i] = 10 + rand() % 90;

}

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

{

arr21[i] = i + 1;

arr22[i] = i + 1;

arr23[i] = i + 1;

}

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

{

arr31[i] = n - i;

arr32[i] = n - i;

arr33[i] = n - i;

}

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

{

if (i< n / 2) {

arr41[i] = i + 1;

arr42[i] = i + 1;

arr43[i] = i + 1;

}

else {

arr41[i] = n - i;

arr42[i] = n - i;

arr43[i] = n - i;

}

}

cout<<"кол-во элмементов: "<< n <<endl;

cout<<setw(15) <<" "<<setw(15) <<"shell"<<setw(15) <<"qs"<<setw(15) <<"qsort";

///////////////////////////////////

double time1 = clock();

shell(arr11, n);

double time2 = clock();

cout<<endl<<setw(15) <<"Рандом"<<setw(15) << (time2 - time1) / CLOCKS\_PER\_SEC;

time1 = clock();

qs(arr12, 0, n - 1);

time2 = clock();

cout<<setw(15) << (time2 - time1) / CLOCKS\_PER\_SEC;

time1 = clock();

qsort(arr13, n, sizeof(int), compare);

time2 = clock();

cout<<setw(15) << (time2 - time1) / CLOCKS\_PER\_SEC;

//////////////////////////////////////

time1 = clock();

shell(arr21, n);

time2 = clock();

cout<<endl<<setw(15) <<"Вызростание"<<setw(15) << (time2 - time1) / CLOCKS\_PER\_SEC;

time1 = clock();

qs(arr22, 0, n - 1);

time2 = clock();

cout<<setw(15) << (time2 - time1) / CLOCKS\_PER\_SEC;

time1 = clock();

qsort(arr23, n, sizeof(int), compare);

time2 = clock();

cout<<setw(15) << (time2 - time1) / CLOCKS\_PER\_SEC;

//////////////////////////////////////

time1 = clock();

shell(arr31, n);

time2 = clock();

cout<<endl<<setw(15) <<"Убывание"<<setw(15) << (time2 - time1) / CLOCKS\_PER\_SEC;

time1 = clock();

qs(arr32, 0, n - 1);

time2 = clock();

cout<<setw(15) << (time2 - time1) / CLOCKS\_PER\_SEC;

time1 = clock();

qsort(arr33, n, sizeof(int), compare);

time2 = clock();

cout<<setw(15) << (time2 - time1) / CLOCKS\_PER\_SEC;

//////////////////////////////////////

time1 = clock();

shell(arr41, n);

time2 = clock();

cout<<endl<<setw(15) <<"Возраст/убыван"<<setw(15) << (time2 - time1) / CLOCKS\_PER\_SEC;

time1 = clock();

qs(arr42, 0, n - 1);

time2 = clock();

cout<<setw(15) << (time2 - time1) / CLOCKS\_PER\_SEC;

time1 = clock();

qsort(arr43, n, sizeof(int), compare);

time2 = clock();

cout<<setw(15) << (time2 - time1) / CLOCKS\_PER\_SEC;

//////////////////////////////////////

}

int main(void)

{

setlocale(0, "");

osnova(100);

cout<<endl<<endl<<endl;

osnova(200);

cout<<endl<<endl<<endl;

osnova(400);

cout<<endl<<endl<<endl;

osnova(1000);

cout<<endl<<endl<<endl;

osnova(2000);

cout<<endl<<endl<<endl;

osnova(4000);

cin.get();

return 0;

}

**Результат работы программы**

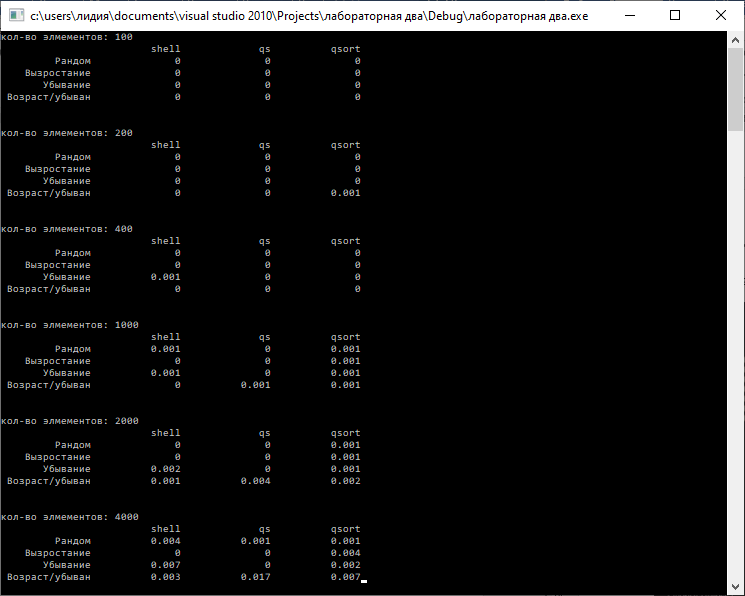


Таблица 1 — График времени работы программы (задание 1)

|  |  |  |
| --- | --- | --- |
| № теста. | Количество элементов. | Время выполнения сортировки в секундах. |
| 1 | 100 | 0.005 |
| 2 | 200 | 0.032 |
| 3 | 400 | 0.379 |
| 4 | 1000 | 6.701 |
| 5 | 2000 | 61.351 |
| 6 | 4000 | 639.104 |
| 7 | 10000 | 6189.73 |
|  |  |  |

### Выводы

В первой части лабораторной работы определили сложность алгоритма, равна O(n­3). А во второй части были разработаны программы с использованием массивов и реализованы алгоритмы сортировки Шелла, быстрой сортировки и qsort. Сортировка Шелла оказалась эффективна на возрастающем массиве, быстрая сортировка и qsort оказались эффективны на всех видах массивов.