Министерство образования РоссийскойФедерации

Пензенский государственный университет

Кафедра «Вычислительная техника»

**ОТЧЕТ**

по лабораторной работе №2

по теме «Оценка времени выполнения программ»

Выполнили:

Студенты группы 21ВВ2

Козлова К.С.

Кривенкова В.С.

Принял:

Митрохин М.А.

Юрова О.В.

Пенза 2021

**Цель работы:**

Приобретение и закрепление навыков оценивания времени выполнения алгоритмов.

**Лабораторные работы:**

**Задание 1:**

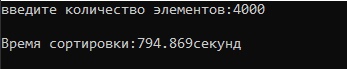
1. Вычислить порядок сложности программы (*О*-символику).
2. Оценить время выполнения программы и кода, выполняющего перемножение матриц, используя функции библиотеки time.h для матриц размерами от 100, 200, 400, 1000, 2000, 4000, 10000.
3. Построить график зависимости времени выполнения программы от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

**Результат работы программы:**

**Задание 1:**

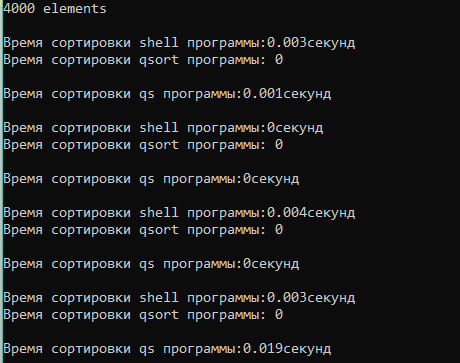


Программа на выходе выдает время выполнения программы и кода, выполняющего перемножение матриц.

Порядок сложности программы: O(n) + O(n\*logn) + O(2\*n^3) есть сложность О (n^3)

|  |  |  |
| --- | --- | --- |
| № теста. | Количество элементов. | Время выполнения сортировки в секундах. |
| 1 | 400 | 0.229 |
| 2 | 1 000 | 5,179 |
| 3 | 2 000 | 68,759 |
| 4 | 4 000 | 794,869 |
| 5 | 10 000 | 9349,36 |

**Задание 2:**



Программа на выходе выдает время выполнения сортировки 3 видов: shellsort, quicksort, qsort (стандартная функция).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Количество элементов | 1 массив | 2 массив | 3 массив | 4 массив |
| |  |  |  | | --- | --- | --- | | shell | qs | qsort | | |  |  |  | | --- | --- | --- | | shell | qs | qsort | | |  |  |  | | --- | --- | --- | | shell | qs | qsort | | |  |  |  | | --- | --- | --- | | shell | qs | qsort | |
| 100 | |  |  |  | | --- | --- | --- | | 0 | 0 | 0 | | |  |  |  | | --- | --- | --- | | 0 | 0 | 0 | | |  |  |  | | --- | --- | --- | | 0 | 0 | 0 | | |  |  |  | | --- | --- | --- | | 0 | 0,001 | 0 | |
| 200 | |  |  |  | | --- | --- | --- | | 0 | 0 | 0 | | |  |  |  | | --- | --- | --- | | 0 | 0 | 0 | | |  |  |  | | --- | --- | --- | | 0 | 0 | 0 | | |  |  |  | | --- | --- | --- | | 0 | 0 | 0 | |
| 400 | |  |  |  | | --- | --- | --- | | 0 | 0 | 0 | | |  |  |  | | --- | --- | --- | | 0 | 0,001 | 0 | | |  |  |  | | --- | --- | --- | | 0 | 0 | 0 | | |  |  |  | | --- | --- | --- | | 0 | 0 | 0 | |
| 1 000 | |  |  |  | | --- | --- | --- | | 0.001 | 0 | 0 | | |  |  |  | | --- | --- | --- | | 0 | 0 | 0 | | |  |  |  | | --- | --- | --- | | 0,001 | 0 | 0 | | |  |  |  | | --- | --- | --- | | 0 | 0.001 | 0 | |
| 2 000 | |  |  |  | | --- | --- | --- | | 0 | 0 | 0 | | |  |  |  | | --- | --- | --- | | 0 | 0 | 0 | | |  |  |  | | --- | --- | --- | | 0.001 | 0 | 0 | | |  |  |  | | --- | --- | --- | | 0.001 | 0.004 | 0 | |
| 4 000 | |  |  |  | | --- | --- | --- | | 0.003 | 0 | 0.001 | | |  |  |  | | --- | --- | --- | | 0 | 0 | 0 | | |  |  |  | | --- | --- | --- | | 0.004 | 0 | 0 | | |  |  |  | | --- | --- | --- | | 0.003 | 0 | 0.019 | |

**Листинг:**

Lab2.cpp (1задание)

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

#include<ctime>

#include <locale.h>

#include <iostream>

using namespace std;

int main(void)

{

setlocale(LC\_ALL, "rus");

setvbuf(stdin, NULL, \_IONBF, 0);

setvbuf(stdout, NULL, \_IONBF, 0);

//clock\_t start, end; // объявляем переменные для определения времени выполнения

int i = 0, j = 0, r, n;

printf("введите количество элементов:");

scanf\_s("%d",&n);

int elem\_c;

int\*\* c = (int\*\*)malloc(n \* sizeof(int));

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

{

c[i] = (int\*)malloc(n \* sizeof(int));

}

srand(time(NULL)); // инициализируем параметры генератора случайных чисел

int\*\* a = (int\*\*)malloc(n \* sizeof(int));

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

{

a[i] = (int\*)malloc(n \* sizeof(int));

}

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

{

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

{

a[i][r] = rand() % 51 - 40;

//printf("%-5d ", a[i][j]);

}

}

srand(time(NULL)); // инициализируем параметры генератора случайных чисел

int\*\* b = (int\*\*)malloc(n \* sizeof(int));

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

{

b[r] = (int\*)malloc(n \* sizeof(int));

}

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

{

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

{

b[r][j] = rand() % 51 - 40;

//printf("%-5d ", b[r][j]);

}

}

time\_t start = clock();

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

{

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

{

elem\_c = 0;

for (r = 0; r < n; r++)

{

elem\_c = elem\_c + a[i][r] \* b[r][j];

c[i][j] = elem\_c;

}

}

}

time\_t stop = clock();

cout << endl << "Время сортировки:" << (stop - start) / 1000.0 << "секунд" << endl;

return(0);

}

Lab2.1.cpp (2 задание)

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <iostream>

#include <cstdlib>

#include <locale.h>

#include <stdlib.h>

#include <stdio.h>

#include<time.h>

using namespace std;

void qs(int\* items, int left, int right);

char res[200] = "result.txt";

FILE\* file;

void shell(int\* items, int count)

{

int i, j, gap, k;

int x, a[5];

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

time\_t start = clock();

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;

}

}

time\_t stop = clock();

file = fopen(res, "a+");

cout << endl << "Время сортировки shell программы:" << (stop - start) / 1000.0 << "секунд" << endl;

fprintf(file, "\nВремя сортировки shell программы:%lf", (double)(stop - start) / CLOCKS\_PER\_SEC);

}

int compare(const void\* x1, const void\* x2)

{

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

}

void osnov(int n) {

time\_t time1, time2;

int\* arr11 = new int[n];

int\* arr12 = new int[n];

int\* arr13 = new int[n];

int\* arr21 = new int[n];

int\* arr22 = new int[n];

int\* arr23 = new int[n];

int\* arr31 = new int[n];

int\* arr32 = new int[n];

int\* arr33 = new int[n];

int\* arr41 = new int[n];

int\* arr42 = new int[n];

int\* arr43 = new int[n];

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

{

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

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

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

}

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

{

arr21[i] = i + 1;

arr22[i] = i + 1;

arr23[i] = i + 1;

}

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

{

arr31[i] = n - i;

arr32[i] = n - i;

arr33[i] = n - i;

}

for (int i = 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;

}

}

shell(arr11, n);

time\_t start = clock();

qs(arr12, 0, n - 1);

time\_t stop = clock();

time1 = clock();

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

time2 = clock();

file = fopen(res,"a+");

cout << "Время сортировки qsort программы: " << (time2 - time1) / CLOCKS\_PER\_SEC << endl;

fprintf(file, "\nВремя сортировки qsort программы:%lf",(double) (time2 - time1) / CLOCKS\_PER\_SEC);

cout << endl << "Время сортировки qs программы:" << (stop - start) / 1000.0 << "секунд" << endl;

fprintf(file, "\nВремя сортировки qs программы:%lf", (double)(stop - start) / CLOCKS\_PER\_SEC);

shell(arr21, n);

start = clock();

qs(arr22, 0, n - 1);

stop = clock();

time1 = clock();

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

time2 = clock();

cout << "Время сортировки qsort программы: " << (time2 - time1) / CLOCKS\_PER\_SEC << endl;

fprintf(file, "\nВремя сортировки qsort программы:%f", (double)(time2 - time1) / CLOCKS\_PER\_SEC);

cout << endl << "Время сортировки qs программы:" << (stop - start) / 1000.0 << "секунд" << endl;

fprintf(file, "\nВремя сортировки qs программы:%f", (double)(stop - start) / CLOCKS\_PER\_SEC);

shell(arr31, n);

start = clock();

qs(arr32, 0, n - 1);

stop = clock();

time1 = clock();

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

time2 = clock();

cout << "Время сортировки qsort программы: " << (time2 - time1) / CLOCKS\_PER\_SEC << endl;

fprintf(file, "\nВремя сортировки qsort программы:%f", (double)(time2 - time1) / CLOCKS\_PER\_SEC);

cout << endl << "Время сортировки qs программы:" << (stop - start) / 1000.0 << "секунд" << endl;

fprintf(file, "\nВремя сортировки qs программы:%f", (double)(stop - start) / CLOCKS\_PER\_SEC);

shell(arr41, n);

start = clock();

qs(arr42, 0, n - 1);

stop = clock();

time1 = clock();

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

time2 = clock();

cout << "Время сортировки qsort программы: " << (time2 - time1) / CLOCKS\_PER\_SEC << endl;

fprintf(file, "\nВремя сортировки qsort программы:%lf", (double)(time2 - time1) / CLOCKS\_PER\_SEC);

cout << endl << "Время сортировки qs программы:" << (stop - start) /1000.0 << "секунд" << endl;

fprintf(file, "\nВремя сортировки qs программы:%lf", (double)(stop - start) / CLOCKS\_PER\_SEC);

fclose(file);

delete[] arr11;

}

int main(void)

{

setlocale(LC\_ALL, "rus");

cout<<endl << "100 elements" << endl;

osnov(100);

cout << endl << "200 elements" << endl;

osnov(200);

cout << endl << "400 elements" << endl;

osnov(400);

cout << endl << "1000 elements" << endl;

osnov(1000);

cout << endl << "2000 elements" << endl;

osnov(2000);

cout << endl << "4000 elements" << endl;

osnov(4000);

}

void qs(int items[], int left, int right) //вызов функции: qs(items, 0, count-1);

{

int i, j;

int 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);

//delete[] items;

//cout << endl << "Время сортировки qs программы:" << (stop - start) / 1000.0 << "секунд" << endl;

}

**Вывод:** Мы приобрели и закрепили навыки оценивания времени выполнения алгоритмов.