Министерство образования РоссийскойФедерации

Пензенский государственный университет

Кафедра «Вычислительная техника»

**ОТЧЕТ**

по лабораторной работе №2

по теме «Оценка времени выполнения программ»

Выполнили:

Студенты группы 21ВВ2

Кожняков Д.М.

Кабачек Н.А.

Принял:

Митрохин М.А.

Юрова О.В.

Пенза 2021

**Цель работы:**

Приобретение и закрепление навыков оценивания времени выполнения алгоритмов.

**Лабораторные работы:**

**Задание 1:**

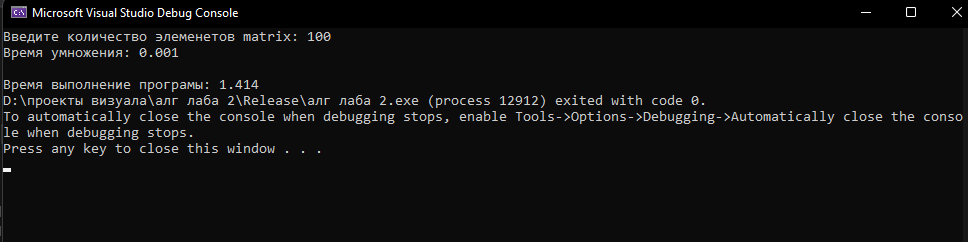
1. Вычислить порядок сложности программы (*О*-символику).
2. Оценить время выполнения программы и кода, выполняющего перемножение матриц, используя функции библиотеки time.h для матриц размерами от 100, 200, 400, 1000, 2000, 4000, 10000.
3. Построить график зависимости времени выполнения программы от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

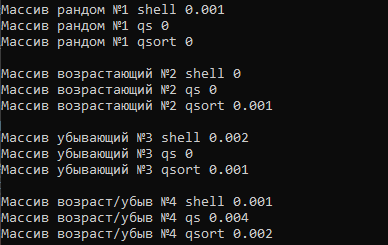
Порядок сложности программы: O(n) + O(n\*logn) + O(2\*n^3) есть сложность О (n^3)

|  |  |  |
| --- | --- | --- |
| № теста. | Количество элементов. | Время выполнения сортировки в секундах. |
| 1 | 100 | 0.001 |
| 2 | 200 | 0.019 |
| 3 | 400 | 0.127 |
| 4 | 1 000 | 13.485 |
| 5 | 2 000 | 74.68 |

Таблица

|  |  |  |
| --- | --- | --- |
| № теста. | Количество элементов. | О (n^3)\*0.000000002 |
| 1 | 100 | 0.002 |
| 2 | 200 | 0.016 |
| 3 | 400 | 0.128 |
| 4 | 1 000 | 2 |
| 5 | 2 000 | 16 |

**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 | 0.001 | | |  |  |  | | --- | --- | --- | | 0 | 0 | 0 | |
| 200 | |  |  |  | | --- | --- | --- | | 0 | 0 | 0 | | |  |  |  | | --- | --- | --- | | 0 | 0 | 0 | | |  |  |  | | --- | --- | --- | | 0 | 0 | 0 | | |  |  |  | | --- | --- | --- | | 0 | 0 | 0 | |
| 400 | |  |  |  | | --- | --- | --- | | 0 | 0 | 0 | | |  |  |  | | --- | --- | --- | | 0 | 0 | 0 | | |  |  |  | | --- | --- | --- | | 0 | 0 | 0 | | |  |  |  | | --- | --- | --- | | 0 | 0 | 0 | |
| 1 000 | |  |  |  | | --- | --- | --- | | 0 | 0 | 0 | | |  |  |  | | --- | --- | --- | | 0 | 0 | 0 | | |  |  |  | | --- | --- | --- | | 0 | 0 | 0.001 | | |  |  |  | | --- | --- | --- | | 0 | 0 | 0 | |
| 2 000 | |  |  |  | | --- | --- | --- | | 0 | 0 | 0 | | |  |  |  | | --- | --- | --- | | 0 | 0 | 0.001 | | |  |  |  | | --- | --- | --- | | 0 | 0.001 | 0 | | |  |  |  | | --- | --- | --- | | 0 | 0.001 | 0 | |
| 4 000 | |  |  |  | | --- | --- | --- | | 0.001 | 0 | 0 | | |  |  |  | | --- | --- | --- | | 0 | 0 | 0.001 | | |  |  |  | | --- | --- | --- | | 0.002 | 0 | 0.001 | | |  |  |  | | --- | --- | --- | | 0.001 | 0.004 | 0.002 | |
| 10 000 | |  |  |  | | --- | --- | --- | | 0.005 | 0.001 | 0.002 | | |  |  |  | | --- | --- | --- | | 0.001 | 0.001 | 0.002 | | |  |  |  | | --- | --- | --- | | 0.01 | 0.001 | 0.002 | | |  |  |  | | --- | --- | --- | | 0.006 | 0.012 | 0.009 | |

**Листинг:**

алг лаб2.cpp (1задание)

#include <iostream>

#include <ctime>

using namespace std;

int main(void)

{

setlocale(0, "");

srand(time(NULL));

int num;

cout << "Введите количество элеменетов matrix: ";

cin >> num;

int k = num;

int\*\* a = new int\* [num];

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

{

a[i] = new int[k];

}

int\*\* b = new int\* [num];

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

{

b[i] = new int[k];

}

int\*\* c = new int\* [num];

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

{

c[i] = new int[k];

}

for (auto i = 0; i < num; i++)

for (auto j = 0; j < num; j++)

a[i][j] = 1 + rand() % 5;

for (auto i = 0; i < num; i++)

for (auto j = 0; j < num; j++)

b[i][j] = 1 + rand() % 5;

double time1 = clock();

for (auto i = 0; i < num; i++) {

for (auto j = 0; j < num; j++) {

c[i][j] = 0;

for (auto y = 0; y < num; y++) {

c[i][j] += a[i][y] \* b[y][j];

}

}

}

double time2 = clock();

cout << "Время умножения: " << (time2 - time1) / 1000.0 << endl;

cout << endl << "Время выполнение програмы: " << clock() / 1000.0;

cin.get();

return 0;

}

алг лаб2.1.cpp (2 задание)

#include <iostream>

#include <cstdlib>

using namespace std;

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;

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;

}

}

}

void qs(int\* items, long int left, long int right) //вызов функции: qs(items, 0, count-1);

{

long int i, j;

long 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);

}

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

{

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

}

int main(void)

{

setlocale(0, "");

const int n = 4000;

int a1[n], a2[n], a3[n],

a4[n], a5[n], a6[n],

a7[n], a8[n], a9[n],

a10[n], a11[n], a12[n];

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

{

a1[i] = 10 + rand() % 90;

a2[i] = 10 + rand() % 90;

a3[i] = 10 + rand() % 90;

}

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

{

a4[i] = i + 1;

a5[i] = i + 1;

a6[i] = i + 1;

}

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

{

a7[i] = n - i;

a8[i] = n - i;

a9[i] = n - i;

}

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

{

if (i < n / 2) {

a10[i] = i + 1;

a11[i] = i + 1;

a12[i] = i + 1;

}

else {

a10[i] = n - i;

a11[i] = n - i;

a12[i] = n - i;

}

}

double time1 = clock();

shell(a1, n);

double time2 = clock();

cout << endl << "Массив рандом №1 shell " << (time2 - time1) / CLOCKS\_PER\_SEC << endl;

time1 = clock();

qs(a2, 0, n - 1);

time2 = clock();

cout << "Массив рандом №1 qs " << (time2 - time1) / CLOCKS\_PER\_SEC << endl;

time1 = clock();

qsort(a3, n, sizeof(int), compare);

time2 = clock();

cout << "Массив рандом №1 qsort " << (time2 - time1) / CLOCKS\_PER\_SEC << endl;

time1 = clock();

shell(a4, n);

time2 = clock();

cout << endl << "Массив возрастающий №2 shell " << (time2 - time1) / CLOCKS\_PER\_SEC << endl;

time1 = clock();

qs(a5, 0, n - 1);

time2 = clock();

cout << "Массив возрастающий №2 qs " << (time2 - time1) / CLOCKS\_PER\_SEC << endl;

time1 = clock();

qsort(a6, n, sizeof(int), compare);

time2 = clock();

cout << "Массив возрастающий №2 qsort " << (time2 - time1) / CLOCKS\_PER\_SEC << endl;

time1 = clock();

shell(a7, n);

time2 = clock();

cout << endl << "Массив убывающий №3 shell " << (time2 - time1) / CLOCKS\_PER\_SEC << endl;

time1 = clock();

qs(a8, 0, n - 1);

time2 = clock();

cout << "Массив убывающий №3 qs " << (time2 - time1) / CLOCKS\_PER\_SEC << endl;

time1 = clock();

qsort(a9, n, sizeof(int), compare);

time2 = clock();

cout << "Массив убывающий №3 qsort " << (time2 - time1) / CLOCKS\_PER\_SEC << endl;

time1 = clock();

shell(a10, n);

time2 = clock();

cout << endl << "Массив возраст/убыв №4 shell " << (time2 - time1) / CLOCKS\_PER\_SEC << endl;

time1 = clock();

qs(a11, 0, n - 1);

time2 = clock();

cout << "Массив возраст/убыв №4 qs " << (time2 - time1) / CLOCKS\_PER\_SEC << endl;

time1 = clock();

qsort(a12, n, sizeof(int), compare);

time2 = clock();

cout << "Массив возраст/убыв №4 qsort " << (time2 - time1) / CLOCKS\_PER\_SEC << endl;

return 0;

}

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