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»

Выполнил:

студент группы 20ВВ4

Айкашев М.А.

Приняли:

Юрова О.В.

Акифьев И.В.

Пенза 2021

**Цель работы:** оценить время выполнения программ.

**Результат выполнения работы.**

**Листинг:**

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <malloc.h>

#include <iostream>

#include <string>

#include <cstdio>

#include <time.h>

using namespace std;

int main()

{

srand(time(NULL));

const int M = 10000;

const int N = 10000;

int\*\* a = NULL;

int\*\* b = NULL;

int\*\* c = NULL;

int i, j;

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

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

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

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

{

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

}

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

{

b[i] = (int\*)malloc(N \* sizeof(int));

}

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

{

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

}

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

{

for (j = 0; j < M; j++) {

a[i][j] = rand() % 100;

b[i][j] = rand() % 100;

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

cout << c[i][j];

}

cout << endl << endl << endl << endl;

}

}

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <malloc.h>

#include <iostream>

#include <string>

#include <cstdio>

#include <time.h>

using namespace std;

int i, j, n, d, c, first, last;

int shell(int A[], int n)

{

d = n;

d = d / 2;

while (d > 0)

{

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

{

j = i;

while (j >= 0 && A[j] > A[j + d])

{

c = A[j];

A[j] = A[j + d];

A[j + d] = c;

j--;

}

}

d = d / 2;

}

cout<< endl << "Итоговый массив после сортировки Шелла: ";

for (i = 0; i < n; i++) cout << A[i] << " "; cout << endl << endl; //вывод массива

return 0;

}

int comp(const void\* a, const void\* b)

{

const int\* x = (int\*)a;

const int\* y = (int\*)b;

if (\*x > \*y) return 1;

else if (\*x < \*y) return -1;

return 0;

}

int quick(int A[], int first, int last)

{

int mid, count;

int f = first, l = last;

mid = A[(f + l) / 2]; //вычисление опорного элемента

do

{

while (A[f]< mid) f++;

while (A[l] > mid) l--;

if (f <= l) //перестановка

{

count = A[f];

A[f] = A[l];

A[l] = count;

f++; l--;

}

} while (f < l);

if (first < l) quick(A, first, l);

if (f < last) quick(A, f, last);

return 0;

}

int main()

{

setlocale(LC\_ALL, "Rus");

srand(time(NULL));

n = 500;

int\* A = new int[n];

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

{

A[i] = rand() % 10;

cout << A[i] << " ";

}

cout << endl;

shell(A, n);

cout << endl;

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

{

A[i] = rand() % 10;

cout << A[i] << " ";

}

cout << endl;

qsort(A, n, sizeof(int), comp);

cout << endl << "Итоговый массив после сортировки qsort: ";

for (i = 0; i < n; i++) cout << A[i] << " "; cout << endl;

cout << endl;

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

{

A[i] = rand() % 10;

cout << A[i] << " ";

}

cout << endl;

first = 0; last = n - 1;

quick(A, first, last);

cout << endl << "Итоговый массив после быстрйо сортировки: ";

for (i = 0; i < n; i++) cout << A[i] << " "; cout << endl;

}

**Задание 1.**

1. Вычислить порядок сложности программы (*О*-символику).

В функции 3 последовательных цикла «for», каждый из которых проходит массив длинной n, значит сложность будет равна: O(n3).

1. 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.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Размер | 100 | 200 | 400 | 1000 | 2000 | 4000 | 10000 |
| Время | 3 c | 4c | 35c | 72c | 780c | 1620c | 2760c |

1. 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 массиве.

|  |  |  |  |
| --- | --- | --- | --- |
| **Алгоритм**  **Наб. дан.** | **Шелла** | **qsort** | **Быстрая сортировка** |
| **Rand** | **0,5c** | **0,5c** | **0,5c** |
| **Возраст.** | **0.1с** | **0.1с** | **0.1с** |
| **Убыв.** | **0,2c** | **0,2c** | **0,2c** |
| **50/50** | **0,3c** | **0,3c** | **0,3c** |

**Вывод:** мы научились пользоваться массивами и структурами, сортировать массива, а также работать со случайными числами. Экспериментально показали правильность расчета сложности программы.