

Сложность О(N^3)

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

#include <locale.h>

#define q 8000

#define w 8000

#pragma comment(linker, "/STACK:615246573000")

int main(void)

{

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

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

setlocale(LC\_ALL, "RUS");

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

int i = 0, j = 0, r;

int a[q][w], b[q][w], c[q][w], elem\_c;

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

while (i < q)

{

while (j < w)

{

a[i][j] = rand() % 100 + 1; // заполняем массив случайными числами

j++;

}

i++;

}

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

i = 0; j = 0;

while (i < q)

{

while (j < w)

{

b[i][j] = rand() % 100 + 1; // заполняем массив случайными числами

j++;

}

i++;

}

start = clock(); // Начало отсчета времени

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

{

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

{

elem\_c = 0;

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

{

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

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

}

}

}

end = clock() - start; // Конец отсчета времени

printf("Размер матрицы %dx%d\n\n", q, w);

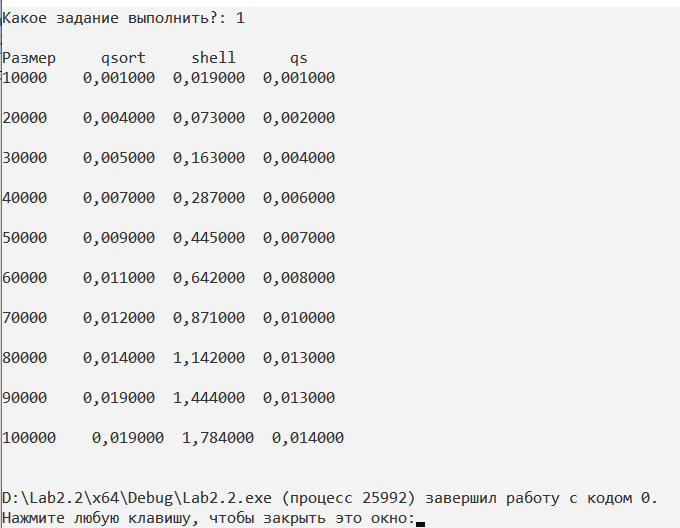
printf("Время работы: %lf", (double)end / CLOCKS\_PER\_SEC);

return(0);

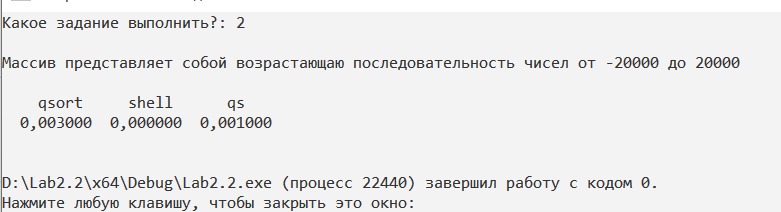
}

2.2

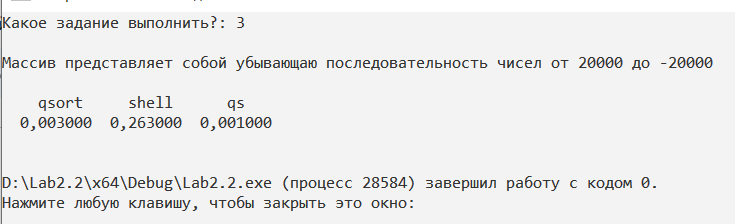
№1



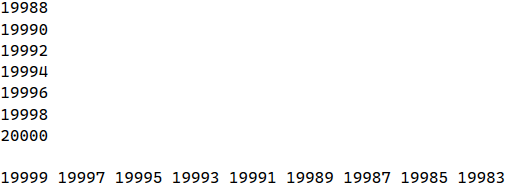
№2

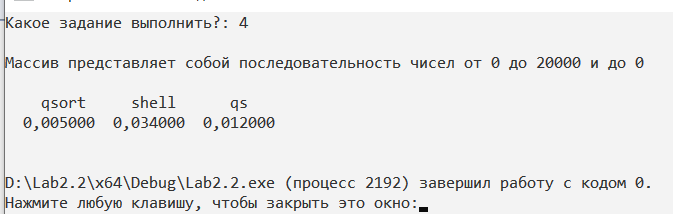


№3



№4





#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <locale.h>

#include <stdlib.h>

#include <time.h>

void shell(int\* items, int count)

{

clock\_t start\_s, end\_s;

int i, j, gap, k;

int x, a[5];

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

start\_s = clock(); // Начало отсчета времени работы функции shell

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;

}

}

end\_s = clock() - start\_s; // Конец отсчета времени работы функции shell

printf("%lf", (double)end\_s / CLOCKS\_PER\_SEC); // Вывод времени работы функции shell

}

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);

}

// Функция для работы qsort

int comp(const void\* i1, const void\* i2) {

return (\*(int\*)i1 - \*(int\*)i2);

}

// Функция для работы с файлами и запуска сортировки

void work\_w\_file(char name1[14]) {

int\* items; // Указатель

int count = 0; // Счетчик

clock\_t start\_qs, end\_qs; // Время работы qs

// Сохранение длины файла

FILE\* file\_1 = fopen(name1, "r");

fseek(file\_1, 0, SEEK\_SET);

while (true)

{

int val;

if (fscanf(file\_1, "%d", &val) == 1)

{

count++;

}

if (feof(file\_1))

{

break;

}

}

fclose(file\_1);

//

items = (int\*)malloc(count \* sizeof(int)); // Расширение в памяти

// Запись в массив из файла множества чисел

file\_1 = fopen(name1, "r");

fseek(file\_1, 0, SEEK\_SET);

for (int i = 0; i < count; ++i) {

fscanf(file\_1, "%d", &(\*(items + i)));

}

fclose(file\_1);

//

clock\_t start = clock(), end; // Начало отсчета времени qsort

qsort(items, count, sizeof(int), comp); // Вызов функции qsort

end = clock() - start; // Время работы qsort

printf(" %lf ", (double)end / CLOCKS\_PER\_SEC); // Вывод времени работы qsort

file\_1 = fopen(name1, "r");

fseek(file\_1, 0, SEEK\_SET);

for (int i = 0; i < count; ++i) {

fscanf(file\_1, "%d", &(\*(items + i)));

}

fclose(file\_1);

shell(items, count); // Вызов функции shell

file\_1 = fopen(name1, "r");

fseek(file\_1, 0, SEEK\_SET);

for (int i = 0; i < count; ++i) {

fscanf(file\_1, "%d", &(\*(items + i)));

}

fclose(file\_1);

start\_qs = clock(); // Начало отсчета времени работы qs

qs(items, 1, count - 1); // Вызов функции qsort

end\_qs = clock() - start\_qs; // Конец отсчета времени работы qs

printf(" %lf\n\n", (double)end\_qs / CLOCKS\_PER\_SEC); // Вывод времени работы qsort

free(items); // Освобождение памяти

}

int main(clock\_t end\_s)

{

setlocale(LC\_ALL, "RUS"); // Подключение русского языка

srand(time(NULL)); // Подключение случайных чисел

clock\_t start\_qs, end\_qs; // Время работы qs

int what; // Для определения вида задачи

printf("Какое задание выполнить?: ");

scanf("%d", &what); // Ввод вида задачи

// 1 Задание

if (what == 1)

{

int\* items;

printf("\nРазмер\t qsort shell qs\n");

for (int count = 10000; count <= 100000; count += 10000) { // Варианты размера массива от 10000 до 100000 с шагом в 10000

printf("%d ", count);

items = (int\*)malloc(count \* sizeof(int)); // Расширение памяти

// Заполнение случайными числами

for (int i = 0; i < count; i++) {

\*(items + i) = rand() % 100;

}

clock\_t start = clock(), end; // Начало отсчета времени работы qsort

qsort(items, count, sizeof(int), comp); // Вызов qsort

end = clock() - start; // Конец отсчета времени работы qsort

printf(" %lf ", (double)end / CLOCKS\_PER\_SEC); // Вывод времени работы qsort

for (int i = 0; i < count; i++) {

\*(items + i) = rand() % 100;

}

shell(items, count); // Вызов shell

for (int i = 0; i < count; i++) {

\*(items + i) = rand() % 100;

}

start\_qs = clock(); // Начало отсчета времени работы qs

qs(items, 0, count - 1); // Вызов qs

end\_qs = clock() - start\_qs; // Конец отсчета времени работы qs

printf(" %lf\n\n", (double)end\_qs / CLOCKS\_PER\_SEC); // Вывод времени работы qs

}

free(items); // Освобождение памяти

}

// 2 задание

if (what == 2)

{

printf("\nМассив представляет собой возрастающую последовательность чисел от -20000 до 20000\n\n qsort shell qs\n");

char name1[14] = "HeightMas.txt";

work\_w\_file(name1); // Вызов work\_w\_file

}

// 3 задание

if (what == 3)

{

printf("\nМассив представляет собой убывающую последовательность чисел от 20000 до -20000\n\n qsort shell qs\n");

char name1[14] = "DownMas.txt";

work\_w\_file(name1); // Вызов work\_w\_file

}

// 4 задание

if (what == 4)

{

printf("\nМассив представляет собой последовательность чисел от 0 до 20000 и до 0\n\n qsort shell qs\n");

char name1[14] = "HeDowMas.txt";

work\_w\_file(name1); // Вызов work\_w\_file

}

return 0;

}