**Московский государственный технический**

**университет им. Н.Э. Баумана**

Факультет «Информатика и управление»

Кафедра ИУ5. Курс «Основы информатики»

Отчет по лабораторной работе №5

# «Сортировка одномерного числового массива»

|  |  |  |
| --- | --- | --- |
| Выполнил: |  | Проверил: |
| студент группы ИУ5-11 |  | преподаватель каф. ИУ5 |
| Анцифров Никита |  | Аксёнова М.В. |
|  |  |  |
| Подпись и дата: |  | Подпись и дата: |

Москва, 2019 г.

**Постановка задачи**

1. Отсортировать числовой массив методом выбора максимального (минимального) элемента и методом пузырькового всплытия.
2. По окончании сортировки вывести отсортированный массив и количество сделанных сравнений и перестановок элементов.
3. Сравнить быстродействие алгоритмов, которое определяется числом сравнений и перестановок, для исходного не отсортированного массива и для исходного массива, отсортированного в прямом и обратном порядке.
4. Исследовать зависимость быстродействия от размера массива. Возможность изменения длины массива реализовать с помощью динамического массива, а для его инициализации использовать датчик случайных чисел.

**Описание входных, выходных и вспомогательных данных**

int const **n** – значение размера статического массива

int **mas1[n]** – массив размера **n**

int **mas2[n]** – массив размера **n**

int **amount\_of\_s** – количество сравнений

int **amount\_of\_p** – количество перестановок

int **i** – шаг внутри цикла

int **j** – шаг внутри цикла

int **k** – шаг внутри цикла

int **r** – значение размера динамического массива

int\* **t1** – динамический массив размера **r**

int\* **t2** – динамический массив размера **r**

int **num** – номер элемента

int **minmax** – найденное минимальное/максимальное значение

int **c** – дополнительная перменная

**Описание прототипов функций**

int **int\_random**(int, int);- функция принимает значения границ интервала и возвращает значение случайного числа в интервале (a,b)

void **mas\_out**(int\*, int);- функция принимает адрес на массив и его размер и выводит этот массив на экран

void **mas\_minmax\_v**(int\*, int\*, int\*, int); - функция принимает адрес на массив, адреса на переменные-счетчики и размер массива и сортирует этот массив методом выбора минимального элемента по возрастанию

void **mas\_minmax\_u**(int\*, int\*, int\*, int); - функция принимает адрес на массив, адреса на переменные-счетчики и размер массива и сортирует этот массив методом выбора максимального элемента по убыванию

void **mas\_bubble\_v**(int\*, int\*, int\*, int); - функция принимает адрес на массив, адреса на переменные-счетчики и размер массива и сортирует этот массив методом пузырька по возрастанию

void **mas\_bubble\_u**(int\*, int\*, int\*, int); - функция принимает адрес на массив, адреса на переменные-счетчики и размер массива и сортирует этот массив методом пузырька по убыванию

void **sp\_out**(int\*, int\*); - функция принимает адреса на переменные-счетчики и выводит эти переменные на экран

**Алгоритм**

Функция **main**





Функции **mas\_minmax\_v** и **mas\_minmax\_u**



Функции **mas\_bubble\_v** и **mas\_bubble\_u**



Функция **int\_random**



**Текст программы**

*L5.cpp*

#include <iostream>

#include <stdlib.h>

#include <time.h>

#include "func\_proto.h"

using namespace std;

int main()

{

setlocale(LC\_ALL, "Russian");

int const n=4;

int mas1[n], mas2[n], amount\_of\_s=0, amount\_of\_p=0;

srand((unsigned int)time(NULL));

rand();

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

{

mas1[i] = int\_random(-100,100);

mas2[i] = mas1[i];

}

cout << "Исходный массив: ";

mas\_out (mas1, n);

cout << endl;

mas\_minmax\_v(mas1, &amount\_of\_s, &amount\_of\_p, n);

cout << "Массив, отсортированный методом выбора минимального значения по возрастанию: ";

mas\_out (mas1, n);

sp\_out (&amount\_of\_s, &amount\_of\_p);

amount\_of\_s=0;

amount\_of\_p=0;

mas\_bubble\_v(mas2, &amount\_of\_s, &amount\_of\_p, n);

cout << "Массив, отсортированный методом пузырька по возрастанию: ";

mas\_out (mas2, n);

sp\_out (&amount\_of\_s, &amount\_of\_p);

amount\_of\_s=0;

amount\_of\_p=0;

mas\_minmax\_v(mas1, &amount\_of\_s, &amount\_of\_p, n);

cout << "Повторная сортировка массива методом выбора минимального значения по возрастанию: " << endl;

sp\_out (&amount\_of\_s, &amount\_of\_p);

amount\_of\_s=0;

amount\_of\_p=0;

mas\_bubble\_v(mas2, &amount\_of\_s, &amount\_of\_p, n);

cout << "Повторная сортировка массива методом пузырька по возрастанию: " << endl;

sp\_out (&amount\_of\_s, &amount\_of\_p);

amount\_of\_s=0;

amount\_of\_p=0;

mas\_minmax\_u(mas1, &amount\_of\_s, &amount\_of\_p, n);

cout << "Массив, отсортированный методом выбора максимального значения по убыванию: ";

mas\_out (mas1, n);

sp\_out (&amount\_of\_s, &amount\_of\_p);

amount\_of\_s=0;

amount\_of\_p=0;

mas\_bubble\_u(mas2, &amount\_of\_s, &amount\_of\_p, n);

cout << "Массив, отсортированный методом пузырька по убыванию: ";

mas\_out (mas2, n);

sp\_out (&amount\_of\_s, &amount\_of\_p);

int r=0;

cout << "Введите значение первоначального размера массива для сравнения: ";

cin >> r;

cout << endl;

for (int k=1; k<=3; k++)

{

amount\_of\_s=0;

amount\_of\_p=0;

int\* t1=new int[r];

int\* t2=new int[r];

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

{

t1[j]=int\_random(-100,100);

t2[j]=t1[j];

}

mas\_minmax\_v(t1, &amount\_of\_s, &amount\_of\_p, r);

cout << "Для массива, состоящего из " << r << " элементов и отсортированного методом выбора минимального значения по возрастанию: " << endl;

sp\_out(&amount\_of\_s, &amount\_of\_p);

mas\_bubble\_v(t2, &amount\_of\_s, &amount\_of\_p, r);

cout << "Для массива, состоящего из " << r << " элементов и отсортированного методом пузырька по возрастанию: " << endl;

sp\_out(&amount\_of\_s, &amount\_of\_p);

delete[]t1;

delete[]t2;

r\*=10;

}

system("pause");

return 0;

}

*func.cpp*

#include <iostream>

using namespace std;

int int\_random(int a, int b)

{

return a+(b-a+1)\*rand()/RAND\_MAX;

}

void mas\_out(int\* m, int n)

{

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

{

cout << m[i] << " ";

}

cout << endl;

}

void mas\_minmax\_v(int\* m, int\* s, int\* p, int n)

{

int num=0, minmax=m[0], c=0;

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

{

num=i;

minmax=m[i];

for (int j=(i+1); j<n; j++)

{

if (m[j]<minmax)

{

num=j;

minmax=m[j];

}

(\*s)++;

}

if (num!=i)

{

c=m[i];

m[i]=minmax;

m[num]=c;

(\*p)++;

}

}

}

void mas\_minmax\_u(int\* m, int\* s, int\* p, int n)

{

int num=0, minmax=m[0], c=0;

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

{

num=i;

minmax=m[i];

for (int j=(i+1); j<n; j++)

{

if (m[j]>minmax)

{

num=j;

minmax=m[j];

}

(\*s)++;

}

if (num!=i)

{

c=m[i];

m[i]=minmax;

m[num]=c;

(\*p)++;

}

}

}

void mas\_bubble\_v(int\* m, int\* s, int\* p, int n)

{

int c=0;

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

{

for (int j=0; j<(n-i-1); j++)

{

if (m[j]>m[j+1])

{

c=m[j];

m[j]=m[j+1];

m[j+1]=c;

(\*p)++;

}

(\*s)++;

}

if ((\*p)==0)

{return;}

}

}

void mas\_bubble\_u(int\* m, int\* s, int\* p, int n)

{

int c=0;

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

{

for (int j=0; j<(n-i-1); j++)

{

if (m[j]<m[j+1])

{

c=m[j];

m[j]=m[j+1];

m[j+1]=c;

(\*p)++;

}

(\*s)++;

}

if ((\*p)==0)

{

return;

}

}

}

void sp\_out(int\* s, int\* p)

{

cout << "Количество сравнений: " << \*s << endl;

cout << "Количество перестановок: " << \*p << endl;

cout << endl;

}

*func\_proto.h*

int int\_random(int, int);

void mas\_out(int\*, int);

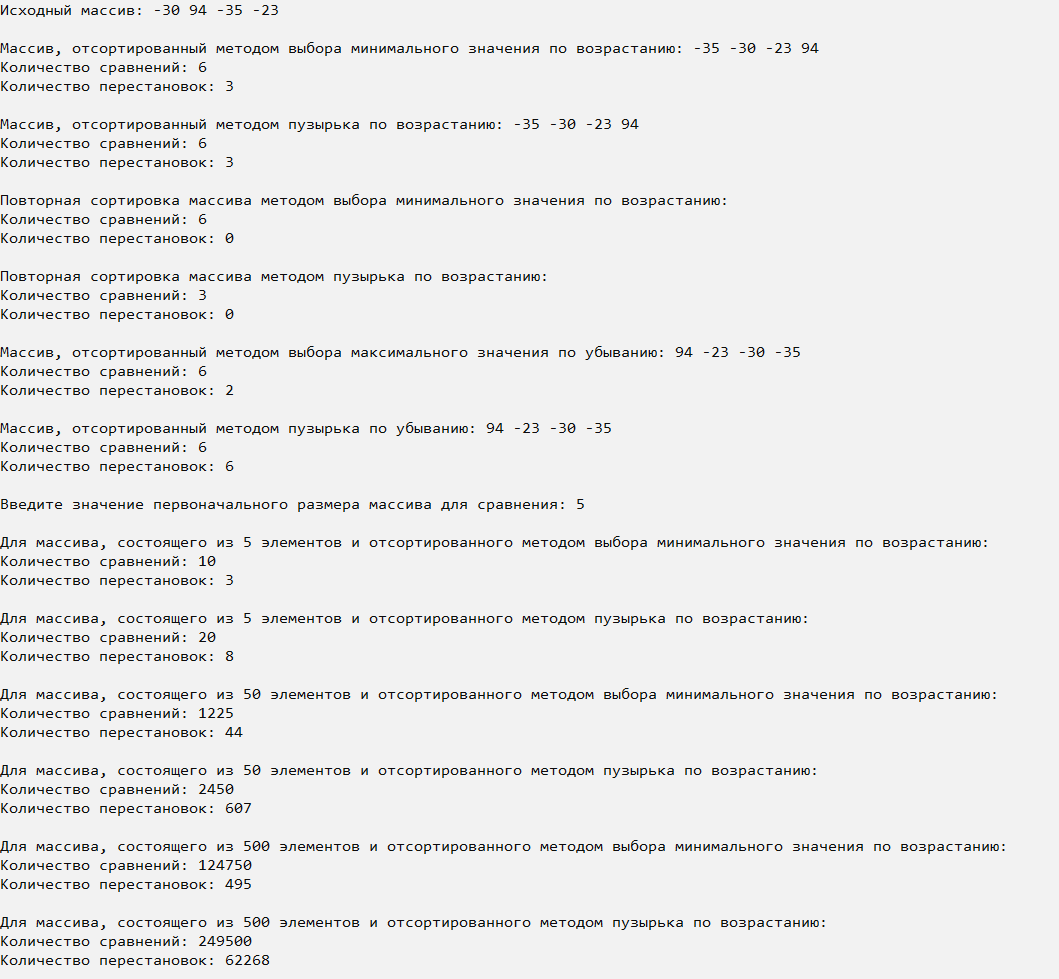
void mas\_minmax\_v(int\*, int\*, int\*, int);

void mas\_minmax\_u(int\*, int\*, int\*, int);

void mas\_bubble\_v(int\*, int\*, int\*, int);

void mas\_bubble\_u(int\*, int\*, int\*, int);

void sp\_out(int\*, int\*);

**Анализ результатов**

Метод выбора минимального/максимального значения эффективнее, чем метод пузырька