**Московский государственный технический**

**университет им. Н.Э. Баумана.**

Факультет «Информатика и управление»

Кафедра ИУ5. Курс «Основы программирования»

Отчет по лабораторной работе №5

«Сортировка одномерного числового массива»

|  |  |  |
| --- | --- | --- |
| Выполнил: |  | Проверил: |
| студент группы ИУ5-12Б  Казицин Алексей |  | преподаватель каф. ИУ5  Козлов А.Д. |
|  |  |  |
| Подпись и дата: |  | Подпись и дата: |

Москва, 2021 г.

Постановка задачи

**Цели работы:**

- освоение методов обработки массивов на примере сортировки массива;

- знакомство с алгоритмами сортировки;

- использование динамических массивов;

-инициализация массивов с помощью датчика случайных чисел;

- оценка быстродействия алгоритма.

**Задание:**

Отсортировать числовой массив методом выбора максимального (минимального) элемента и методом пузырькового всплытия. По окончании сортировки вывести отсортированный массив и количество сделанных сравнений и перестановок элементов.

Сравнить быстродействие алгоритмов, которое определяется числом сравнений и перестановок, для исходного не отсортированного массива и для исходного массива, отсортированного в прямом и обратном порядке.

Исследовать зависимость быстродействия от размера массива. Возможность изменения длины массива реализуйте с помощью динамического массива, а для его инициализации используйте датчик случайных чисел. Результаты исследования выведите в виде отформатированной таблицы.

Разработка алгоритма

**Описание используемых переменных и функций с указанием наименования, типа (int, float, и т.п.) и назначения в программе:**

int ksravn (количество сравнений); int kper (количество перестановок); int i, j (счетчики в циклах); int n (размер массивов); int a[n], int b[n] (использованные массивы); int size (размер массивов); int x[size], y[size] (использованные массивы);

void\_bubble\_vozr(int a[], int n, int \*ksravn, int \*kрег)(функция пузырьковой сортировки массива по возрастанию принимает указатель на первый элемент массива, количество элементов, количество сравнений и перестановок)

void\_bubble\_ub(int a[], int n, int \*ksravn, int \*kрег) - аналогичная функция пузырьковой сортировки по убыванию

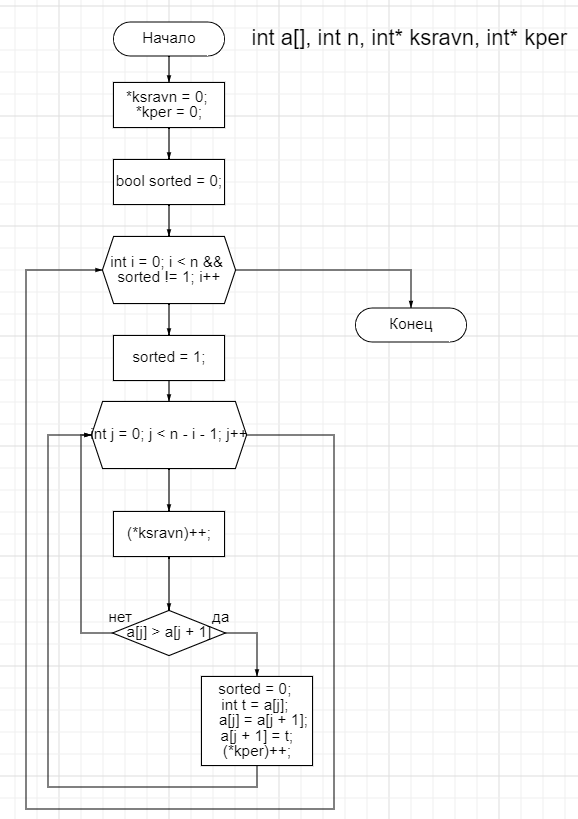
void\_vybor\_vozr(int a[], int n, int \*ksravn, int \*kрег)(функция сортировки методом выбора по возрастанию принимает указатель на первый элемент массива, количество элементов, количество сравнений и перестановок)

void\_vybor\_ub(int a[], int n, int \*ksravn, int \*kрег) - аналогичная функция сортировки методом выбора по убыванию

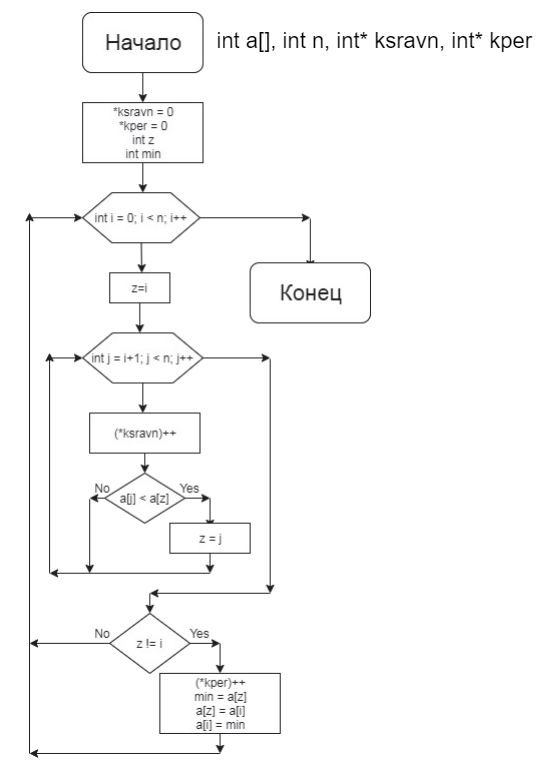
Int iRandom(int a, int b) (функция, возвращающая целое число из полученного [a, b])

double dRandom(int a, int b) – аналогичная функция для вещественных чисел.

Блок-схема сортировки методом пузырька



Блок-схема сортировки методом выбора максимального (минимального) элемента



Текст программы

#include <time.h>

#include <stdlib.h>

#include <iostream>

#include <iomanip>

#include <windows.h>

using namespace std;

int iRandom(int a, int b) {

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

}

double dRandom(int a, int b){

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

}

void bubble\_vozr(int a[], int n, int\* ksravn, int\* kper) {

\*ksravn = 0;

\*kper = 0;

bool sorted = 0;

for (int i = 0; i < n && sorted != 1; i++) {

sorted = 1;

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

(\*ksravn)++;

if (a[j] > a[j + 1]){

sorted = 0;

int t = a[j];

a[j] = a[j + 1];

a[j + 1] = t;

(\*kper)++;

}

}

}

}

void bubble\_ub(int a[], int n, int\* ksravn, int\* kper) {

\*ksravn = 0;

\*kper = 0;

bool sorted = 0;

for (int i = 0; i < n && sorted != 1; i++) {

sorted = 1;

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

(\*ksravn)++;

if (a[j] < a[j + 1]){

sorted = 0;

int t = a[j];

a[j] = a[j + 1];

a[j + 1] = t;

(\*kper)++;

}

}

}

}

void vybor\_vozr(int a[], int n, int\* ksravn, int\* kper) {

\*ksravn = 0;

\*kper = 0;

int z;

int min;

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

z = i;

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

(\*ksravn)++;

if (a[j] < a[z]) {

z = j;

}

}

if (z != i) {

(\*kper)++;

min = a[z];

a[z] = a[i];

a[i] = min;

}

}

}

void vybor\_ub(int a[], int n, int\* ksravn, int\* kper) {

\*ksravn = 0;

\*kper = 0;

int z;

int max;

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

z = i;

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

(\*ksravn)++;

if (a[j] > a[z]) {

z = j;

}

}

(\*kper)++;

max = a[z];

a[z] = a[i];

a[i] = max;

}

}

int main(){

SetConsoleCP(65001);

SetConsoleOutputCP(65001);

int ksravn = 0;

int kper = 0;

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

rand();

int n;

cout << "Введите длину массива:\n";

cin >> n;

cout << endl << endl << "---------------------------ПЕРВАЯ ЧАСТЬ---------------------------" << endl << endl;

int\* a = new int[n];

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

a[i] = iRandom(1, 9);

cout << "a[" << i << "]=" << a[i] << endl;

}

int\* b = new int[n];

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

b[i] = a[i];

}

cout << endl;

vybor\_vozr(a, n, &ksravn, &kper);

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

cout << a[i] << " ";

}

cout << endl;

cout << "1)Метод выбора(не отсортированный): " << endl << "Количество сравнений — " << ksravn << endl << "Количество перестановок — " << kper << endl << endl;

bubble\_vozr(b, n, &ksravn, &kper);

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

cout << b[i] << " ";

}

cout << endl;

cout << "1)Метод пузырька(не отсортированный): " << endl << "Количество сравнений - " << ksravn << endl << "Количество перестановок - " << kper << endl << endl;

vybor\_vozr(a, n, &ksravn, &kper);

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

cout << a[i] << " ";

}

cout << endl;

cout << "2)Метод выбора(отсортированный): " << endl << "Количество сравнений — " << ksravn << endl << "Количество перестановок — " << kper << endl << endl;

bubble\_vozr(b, n, &ksravn, &kper);

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

cout << b[i] << " ";

}

cout << endl;

cout << "2)Метод пузырька(отсортированный): " << endl << "Количество сравнений - " << ksravn << endl << "Количество перестановок - " << kper << endl << endl;

vybor\_ub(a, n, &ksravn, &kper);

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

cout << a[i] << " ";

}

cout << endl;

cout << "3)Метод выбора(отсортированный в обратном порядке): " << endl << "Количество сравнений — " << ksravn << endl << "Количество перестановок — " << kper << endl << endl;

bubble\_ub(b, n, &ksravn, &kper);

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

cout << b[i] << " ";

}

cout << endl;

cout << "3)Метод пузырька(отсортированный в обратном порядке): " << endl << "Количество сравнений — " << ksravn << endl << "Количество перестановок — " << kper << endl << endl;

delete[] a;

delete[] b;

cout << endl << endl << endl << "----------------------------ВТОРАЯ ЧАСТЬ---------------------------" << endl;

cout << "-----Исследование зависимости быстродействия от размера массива----" << endl << endl;

int\* x;

int\* y;

int size = 5;

cout << right << setw(29) << "Сравнения" << setw(32) << "Перестановки" << endl;

while (size <= 500) {

x = new int[size];

y = new int[size];

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

{

x[i] = iRandom(1, 100);

}

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

y[i] = x[i];

}

bubble\_vozr(x, size, &ksravn, &kper);

cout << endl << size << " элементов: " << endl << setw(20) << left << "Пузырек: " << setw(29) << ksravn << left << setw(29) << kper << endl;

vybor\_vozr(y, size, &ksravn, &kper);

cout << setw(20) << left << "Выбор: " << setw(29) << ksravn << left << setw(29) << kper << endl;

delete[] x;

delete[] y;

size \*= 10;

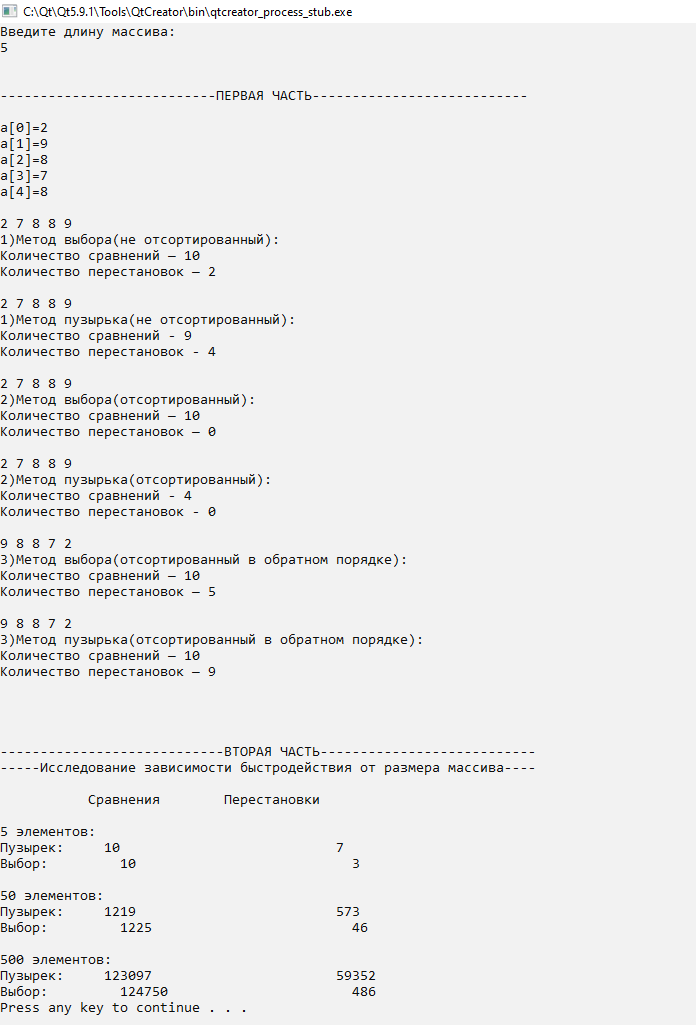
}

system("pause");

return 0;

}

Анализ результатов

Длина массива - 5

Длина массива - 4