

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ  
ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ им. В. Г. ШУХОВА»  
(БГТУ им. В.Г. Шухова)**

Кафедра программного обеспечения вычислительной техники и автоматизированных  
систем

## **Лабораторная работа №3**

по дисциплине: Алгоритмы и структуры данных  
тема: «Сравнительный анализ методов сортировки (С)»

Выполнил: ст. группы ПВ-202  
Буйвало Анастасия Андреевна

Проверил:  
Кабалянец Петр Степанович  
Маньшин Илья Михайлович

Белгород 2021 г.

## Лабораторная работа №3

### «Сравнительный анализ методов сортировки (Pascal/C)»

#### Цель работы:

Изучение методов сортировки массивов и приобретение навыков в проведении сравнительного анализа различных методов сортировки.

#### Задание:

1. Изучить временные характеристики алгоритмов.
2. Изучить методы сортировки:
  1. включением;
  2. выбором;
  3. обменом:
    - а. улучшенная обменом 1;
    - б. улучшенная обменом 2;
  4. Шелла;
  5. Хоара;
  6. пирамидальная.
3. Программно реализовать методы сортировки массивов.
4. Разработать и программно реализовать средство для проведения экспериментов по определению временных характеристик алгоритмов сортировки.
5. Провести эксперименты по определению временных характеристик алгоритмов сортировки. Результаты экспериментов представить в виде таблицы 9, клетки которой содержат количество операций сравнения при выполнении алгоритма сортировки массива с заданным количеством элементов. Провести эксперимент для упорядоченных, неупорядоченных и упорядоченных в обратном порядке массивов (для каждого типа массива заполнить отдельную таблицу).
6. Построить график зависимости количества операций сравнения от количества элементов в сортируемом массиве.
7. Определить аналитическое выражение функции зависимости количества операций сравнения от количества элементов в массиве.
8. Определить порядок функций временной сложности алгоритмов сортировки при сортировке упорядоченных, неупорядоченных и упорядоченных в обратном порядке массивов.

## Выполнение работы:

### Задание 4:

```
#include <stdio.h>
#include <stdlib.h>

#define N 100

//сортировка включением/вставками
// если вставляемый элемент меньше сравниваемого, то продолжаем сравнение
int sort_ins(int a[], int n)
{
    int comps = 0;
    int i, j, k;
    for(j = 1; j < n; j++){
        k = a[j];
        i = j - 1;
        while (++comps && k < a[i] && i >= 0){
            a[i+1] = a[i];
            i -= 1;
        }
        a[i+1] = k;
    }
    return comps;
}

//обмен
void swap(int* a, int* b)
{
    int t = *a;
    *a = *b;
    *b = t;
}

//сортировка выбором(минимальный элемент в оставшейся части)
int sort_sel(int a[], int n)
{
    int comps = 0;
    for(int i = 0; i < n - 1; i++){
        int imin = i;
        for(int j = i + 1; j < n; j++) {
            if (++comps && a[j] < a[imin]) {
                imin = j;
            }
        }
        swap(&a[i], &a[imin]); // обмениваем минимальный элемент с текущим
    }
    return comps;
}

//сортировка пузырьком(обмениваются рядом стоящие элементы, если нарушена
упорядоченность)
int sort_bubble(int a[], int n)
{
    int comps = 0;
    for(int j = 0; j < n-1; j++) {
        for(int i = n-1; i > 0; i--){
            if (++comps && a[i] < a[i - 1])
                swap(&a[i], &a[i - 1]);
        }
    }
}
```

```

        return comps;
    }

//2 версия сортировка пузырьком
// (если ничего не обменялось в текущий проход, то можно выйти досрочно)
int sort_bubble2(int a[], int n)
{
    int comps = 0;
    for(int j = 0; j < n-1; j++) {
        int t = 0;
        for(int i = n-1; i > 0; i--){
            if (++comps && a[i] < a[i - 1]) {
                swap(&a[i], &a[i - 1]);
                t++;
            }
        }
        if(!t)
            break;
    }
    return comps;
}

//3 версия сортировка пузырьком
//(границы по i до j)
int sort_bubble3(int a[], int n)
{
    int comps = 0;
    for(int j = 0; j < n-1; j++) {
        int t = 0;
        for(int i = n-1; i > j; i--){
            if (++comps && a[i] < a[i - 1]) {
                swap(&a[i], &a[i - 1]);
                t++;
            }
        }
        if(!t)
            break;
    }
    return comps;
}

//сортировка шелла
//(сортировка вставками с шагом)
int sort_shell(int a[], int n)
{
    int comps = 0;
    int t = n/2; //вычисляем начальный шаг
    int k, j, i;
    while(t) {
        for(j = t; j < n; j++){
            k = a[j];
            i = j - t;
            while (++comps && k < a[i] && i >= 0){ //сортировка вставками чисел через
шаг t
                a[i+t] = a[i];
                i -= t;
            }
            a[i+t] = k;
        }
        t /= 2; // уменьшение шага
    }
    return comps;
}

```

```

}

//сортировка хоара
//мысленное деление массива по отношению к элементу x(больше, меньше))
int sort_hoar_al(int a [], int l, int r, int n){

    int comps = 0;

    int x = a[l];
    int i = l; //левая граница
    int j = r; //правая граница

    while (i <= j){
        while (++comps && a[i] < x && i < n) //сдвигаем левую границу до первого
числа,
            i++;
        while (++comps && a[j] > x && j > 0) //сдвигаем правую границу до первого
числа,
            j--;
        if (i <= j){
            swap(&a[i], &a[j]); // обмениваем числа
            i++;
            j--;
        }
    }
    if (l < j)
        comps += sort_hoar_al(a, l, j, n); //если текущая правая граница
        // зашла за стартовую левую
    if (i < r)
        comps += sort_hoar_al(a, i, r, n); //если текущая левая
        //зашла за стартовую правую
    return comps;
}

//обертка сортировки хоара
int sort_hoar(int a[], int n)
{
    return sort_hoar_al(a, 0, n-1, n); //изначально левая и правая границы начало и
конец массива
}

int Sift(int A[],int L,int R)
{
    int i,j,x,k;
    i = L;
    j = 2 * L+1;
    x = A[L];
    int comps = 0;
    if ((j < R) && ++comps && (A[j] < A[j+1]))
        j++;
    while ((j <= R) && ++comps && (x < A[j])) {
        k = A[i];
        A[i] = A[j];
        A[j] = k;
        i = j;
        j = 2 * j + 1;
        if ((j < R) && ++comps && (A[j] < A[j + 1]))
            j++;
    }
    return comps;
}

```

```

/* пирамидальная функция сортировки */
int HeapSort(int A[],int nn)
{
    int L,R,x;
    L = nn/2 ;
    R = nn-1;
    int comps = 0;
    while ( L > 0 ) { //Построение пирамиды из исходного массива
        L = L - 1;
        comps += Sift(A,L,R);
    }
    while ( R > 0 ) //Сортировка: пирамида => отсортированный массив
    {
        x = A[0];
        A[0] = A[R];
        A[R] = x;
        R--;
        comps += Sift(A,L,R);
    }
    return comps;
}

void copy_arr(int a[], int b[], int n)
{
    for(int i = 0; i < n; i++)
        b[i] = a[i];
}

void all_sort(int a[], int n)
{
    int b[N];

    copy_arr(a, b, n);
    printf("сравнения в сортировке включением/вставками : %d\n", sort_ins(b, n));

    copy_arr(a, b, n);
    printf("сравнения в сортировке выбором : %d\n", sort_sel(b, n));

    copy_arr(a, b, n);
    printf("сравнения в сортировке обменом : %d\n", sort_bubble(b, n));

    copy_arr(a, b, n);
    printf("сравнения в сортировке обменом 1 : %d\n", sort_bubble2(b, n));

    copy_arr(a, b, n);
    printf("сравнения в сортировке обменом 2 : %d\n", sort_bubble3(b, n));

    copy_arr(a, b, n);
    printf("сравнения в сортировке Шелла : %d\n", sort_shell(b, n));

    copy_arr(a, b, n);
    printf("сравнения в сортировке Хоара : %d\n", sort_hoar(b, n));

    copy_arr(a, b, n);
    printf("сравнения в сортировке Пирамидальная : %d\n", HeapSort(b, n));
    printf("\n");
}

//0-по возрастанию, 1 - по убыванию, 2 - рандом
void generate_arr(int a[], int n, int t)
{
    switch (t) {

```

```

        case 0: {
            for (int i = 0; i < n; i++)
                a[i] = i;
            break;
        }
        case 1: {
            int j = n - 1;
            for(int i = 0; i < n; i++) {
                a[i] = j;
                j--;
            }
            break;
        }
        case 2: {
            srand(2);
            for(int i = 0; i < n; i++)
                a[i] = rand();
            break;
        }
        default:{
            printf("передано неверное значение\n");
            break;
        }
    }
}

//вывод массива a размера n
void output_arr(int a[], int n)
{
    for(int i = 0; i < n; i++)
        printf("%d ", a[i]);
    printf("\n");
}

void condit_sort(int a[], int n)
{
    printf("массив возрастающий(n = %d)\n", n);
    generate_arr(a, n, 0);
    all_sort(a, n);

    printf("массив убывающий(n = %d)\n", n);
    generate_arr(a, n, 1);
    all_sort(a, n);

    printf("массив не упорядочен(n = %d)\n", n);
    generate_arr(a, n, 2);
    all_sort(a, n);
}

int main()
{
    int a[N];
    for(int n = 5; n < 50; n += 5)
        condit_sort(a, n);
}

```

## Результаты работы программы:

массив возрастающий( $n = 35$ )

сравнения в сортировке включением/вставками : 34

сравнения в сортировке выбором : 595

сравнения в сортировке обменом : 1156

сравнения в сортировке обменом 1 : 34

сравнения в сортировке обменом 2 : 34

сравнения в сортировке Шелла : 143

сравнения в сортировке Хоара : 663

сравнения в сортировке Пирамидальная : 266

массив убывающий( $n = 35$ )

сравнения в сортировке включением/вставками : 629

сравнения в сортировке выбором : 595

сравнения в сортировке обменом : 1156

сравнения в сортировке обменом 1 : 1156

сравнения в сортировке обменом 2 : 595

сравнения в сортировке Шелла : 226

сравнения в сортировке Хоара : 680

сравнения в сортировке Пирамидальная : 233

массив не упорядочен( $n = 35$ )

сравнения в сортировке включением/вставками : 295

сравнения в сортировке выбором : 595

сравнения в сортировке обменом : 1156

сравнения в сортировке обменом 1 : 1088

сравнения в сортировке обменом 2 : 592

сравнения в сортировке Шелла : 224

сравнения в сортировке Хоара : 271

сравнения в сортировке Пирамидальная : 257



массив возрастающий( $n = 45$ )

сравнения в сортировке включением/вставками : 44

сравнения в сортировке выбором : 990

сравнения в сортировке обменом : 1936

сравнения в сортировке обменом 1 : 44

сравнения в сортировке обменом 2 : 44

сравнения в сортировке Шелла : 184

сравнения в сортировке Хоара : 1078

сравнения в сортировке Пирамидальная : 378

массив убывающий( $n = 45$ )

сравнения в сортировке включением/вставками : 1034

сравнения в сортировке выбором : 990

сравнения в сортировке обменом : 1936

сравнения в сортировке обменом 1 : 1936

сравнения в сортировке обменом 2 : 990

сравнения в сортировке Шелла : 280

сравнения в сортировке Хоара : 1100

сравнения в сортировке Пирамидальная : 330

массив не упорядочен( $n = 45$ )

сравнения в сортировке включением/вставками : 444

сравнения в сортировке выбором : 990

сравнения в сортировке обменом : 1936

сравнения в сортировке обменом 1 : 1760

сравнения в сортировке обменом 2 : 980

сравнения в сортировке Шелла : 282

сравнения в сортировке Хоара : 377

сравнения в сортировке Пирамидальная : 363

массив возрастающий( $n = 40$ )

сравнения в сортировке включением/вставками : 39

сравнения в сортировке выбором : 780

сравнения в сортировке обменом : 1521

сравнения в сортировке обменом 1 : 39

сравнения в сортировке обменом 2 : 39

сравнения в сортировке Шелла : 162

сравнения в сортировке Хоара : 858

сравнения в сортировке Пирамидальная : 321

массив убывающий( $n = 40$ )

сравнения в сортировке включением/вставками : 819

сравнения в сортировке выбором : 780

сравнения в сортировке обменом : 1521

сравнения в сортировке обменом 1 : 1521

сравнения в сортировке обменом 2 : 780

сравнения в сортировке Шелла : 254

сравнения в сортировке Хоара : 876

сравнения в сортировке Пирамидальная : 278

массив не упорядочен( $n = 40$ )

сравнения в сортировке включением/вставками : 354

сравнения в сортировке выбором : 780

сравнения в сортировке обменом : 1521

сравнения в сортировке обменом 1 : 1404

сравнения в сортировке обменом 2 : 774

сравнения в сортировке Шелла : 275

сравнения в сортировке Хоара : 315

сравнения в сортировке Пирамидальная : 309

массив возрастающий( $n = 30$ )

сравнения в сортировке включением/вставками : 29

сравнения в сортировке выбором : 435

сравнения в сортировке обменом : 841

сравнения в сортировке обменом 1 : 29

сравнения в сортировке обменом 2 : 29

сравнения в сортировке Шелла : 94

сравнения в сортировке Хоара : 493

сравнения в сортировке Пирамидальная : 211

массив убывающий( $n = 30$ )

сравнения в сортировке включением/вставками : 464

сравнения в сортировке выбором : 435

сравнения в сортировке обменом : 841

сравнения в сортировке обменом 1 : 841

сравнения в сортировке обменом 2 : 435

сравнения в сортировке Шелла : 163

сравнения в сортировке Хоара : 506

сравнения в сортировке Пирамидальная : 188

массив не упорядочен( $n = 30$ )

сравнения в сортировке включением/вставками : 227

сравнения в сортировке выбором : 435

сравнения в сортировке обменом : 841

сравнения в сортировке обменом 1 : 783

сравнения в сортировке обменом 2 : 432

сравнения в сортировке Шелла : 156

сравнения в сортировке Хоара : 240

сравнения в сортировке Пирамидальная : 200

массив возрастающий( $n = 25$ )

сравнения в сортировке включением/вставками : 24

сравнения в сортировке выбором : 300

сравнения в сортировке обменом : 576

сравнения в сортировке обменом 1 : 24

сравнения в сортировке обменом 2 : 24

сравнения в сортировке Шелла : 78

сравнения в сортировке Хоара : 348

сравнения в сортировке Пирамидальная : 167

массив убывающий( $n = 25$ )

сравнения в сортировке включением/вставками : 324

сравнения в сортировке выбором : 300

сравнения в сортировке обменом : 576

сравнения в сортировке обменом 1 : 576

сравнения в сортировке обменом 2 : 300

сравнения в сортировке Шелла : 118

сравнения в сортировке Хоара : 360

сравнения в сортировке Пирамидальная : 143

массив не упорядочен( $n = 25$ )

сравнения в сортировке включением/вставками : 156

сравнения в сортировке выбором : 300

сравнения в сортировке обменом : 576

сравнения в сортировке обменом 1 : 528

сравнения в сортировке обменом 2 : 297

сравнения в сортировке Шелла : 124

сравнения в сортировке Хоара : 187

сравнения в сортировке Пирамидальная : 161

массив возрастающий( $n = 20$ )

сравнения в сортировке включением/вставками : 19

сравнения в сортировке выбором : 190

сравнения в сортировке обменом : 361

сравнения в сортировке обменом 1 : 19

сравнения в сортировке обменом 2 : 19

сравнения в сортировке Шелла : 62

сравнения в сортировке Хоара : 228

сравнения в сортировке Пирамидальная : 121

массив убывающий( $n = 20$ )

сравнения в сортировке включением/вставками : 209

сравнения в сортировке выбором : 190

сравнения в сортировке обменом : 361

сравнения в сортировке обменом 1 : 361

сравнения в сортировке обменом 2 : 190

сравнения в сортировке Шелла : 98

сравнения в сортировке Хоара : 236

сравнения в сортировке Пирамидальная : 105

массив не упорядочен( $n = 20$ )

сравнения в сортировке включением/вставками : 120

сравнения в сортировке выбором : 190

сравнения в сортировке обменом : 361

сравнения в сортировке обменом 1 : 323

сравнения в сортировке обменом 2 : 187

сравнения в сортировке Шелла : 89

сравнения в сортировке Хоара : 148

сравнения в сортировке Пирамидальная : 118

массив возрастающий( $n = 15$ )

сравнения в сортировке включением/вставками : 14

сравнения в сортировке выбором : 105

сравнения в сортировке обменом : 196

сравнения в сортировке обменом 1 : 14

сравнения в сортировке обменом 2 : 14

сравнения в сортировке Шелла : 34

сравнения в сортировке Хоара : 133

сравнения в сортировке Пирамидальная : 77

массив убывающий( $n = 15$ )

сравнения в сортировке включением/вставками : 119

сравнения в сортировке выбором : 105

сравнения в сортировке обменом : 196

сравнения в сортировке обменом 1 : 196

сравнения в сортировке обменом 2 : 105

сравнения в сортировке Шелла : 61

сравнения в сортировке Хоара : 140

сравнения в сортировке Пирамидальная : 66

массив не упорядочен( $n = 15$ )

сравнения в сортировке включением/вставками : 64

сравнения в сортировке выбором : 105

сравнения в сортировке обменом : 196

сравнения в сортировке обменом 1 : 168

сравнения в сортировке обменом 2 : 102

сравнения в сортировке Шелла : 68

сравнения в сортировке Хоара : 90

сравнения в сортировке Пирамидальная : 75

массив возрастающий( $n = 10$ )

сравнения в сортировке включением/вставками : 9

сравнения в сортировке выбором : 45

сравнения в сортировке обменом : 81

сравнения в сортировке обменом 1 : 9

сравнения в сортировке обменом 2 : 9

сравнения в сортировке Шелла : 22

сравнения в сортировке Хоара : 63

сравнения в сортировке Пирамидальная : 41

массив убывающий( $n = 10$ )

сравнения в сортировке включением/вставками : 54

сравнения в сортировке выбором : 45

сравнения в сортировке обменом : 81

сравнения в сортировке обменом 1 : 81

сравнения в сортировке обменом 2 : 45

сравнения в сортировке Шелла : 35

сравнения в сортировке Хоара : 66

сравнения в сортировке Пирамидальная : 35

массив не упорядочен( $n = 10$ )

сравнения в сортировке включением/вставками : 25

сравнения в сортировке выбором : 45

сравнения в сортировке обменом : 81

сравнения в сортировке обменом 1 : 63

сравнения в сортировке обменом 2 : 42

сравнения в сортировке Шелла : 32

сравнения в сортировке Хоара : 57

сравнения в сортировке Пирамидальная : 41

массив возрастающий( $n = 5$ )

сравнения в сортировке включением/вставками : 4

сравнения в сортировке выбором : 10

сравнения в сортировке обменом : 16

сравнения в сортировке обменом 1 : 4

сравнения в сортировке обменом 2 : 4

сравнения в сортировке Шелла : 7

сравнения в сортировке Хоара : 18

сравнения в сортировке Пирамидальная : 12

массив убывающий( $n = 5$ )

сравнения в сортировке включением/вставками : 14

сравнения в сортировке выбором : 10

сравнения в сортировке обменом : 16

сравнения в сортировке обменом 1 : 16

сравнения в сортировке обменом 2 : 10

сравнения в сортировке Шелла : 11

сравнения в сортировке Хоара : 20

сравнения в сортировке Пирамидальная : 10

массив не упорядочен( $n = 5$ )

сравнения в сортировке включением/вставками : 7

сравнения в сортировке выбором : 10

сравнения в сортировке обменом : 16

сравнения в сортировке обменом 1 : 8

сравнения в сортировке обменом 2 : 7

сравнения в сортировке Шелла : 10

сравнения в сортировке Хоара : 19

сравнения в сортировке Пирамидальная : 11



### Задание 5:

#### Упорядоченные массивы: (количество сравнений)

Сортировка	Количество элементов в массиве								
	5	10	15	20	25	30	35	40	45
Включением	4	9	14	19	24	29	34	39	44
Выбором	10	45	105	190	300	435	595	780	990
Обменом	20	81	196	361	576	841	1156	1521	1936
Обменом 1	4	9	14	19	24	29	34	39	44
Обменом 2	4	9	14	19	24	29	34	39	44
Шелла	7	22	34	62	78	94	143	162	184
Хоара	18	63	133	228	348	493	663	858	1078
Пирамидальная	12	41	77	121	167	211	266	321	378

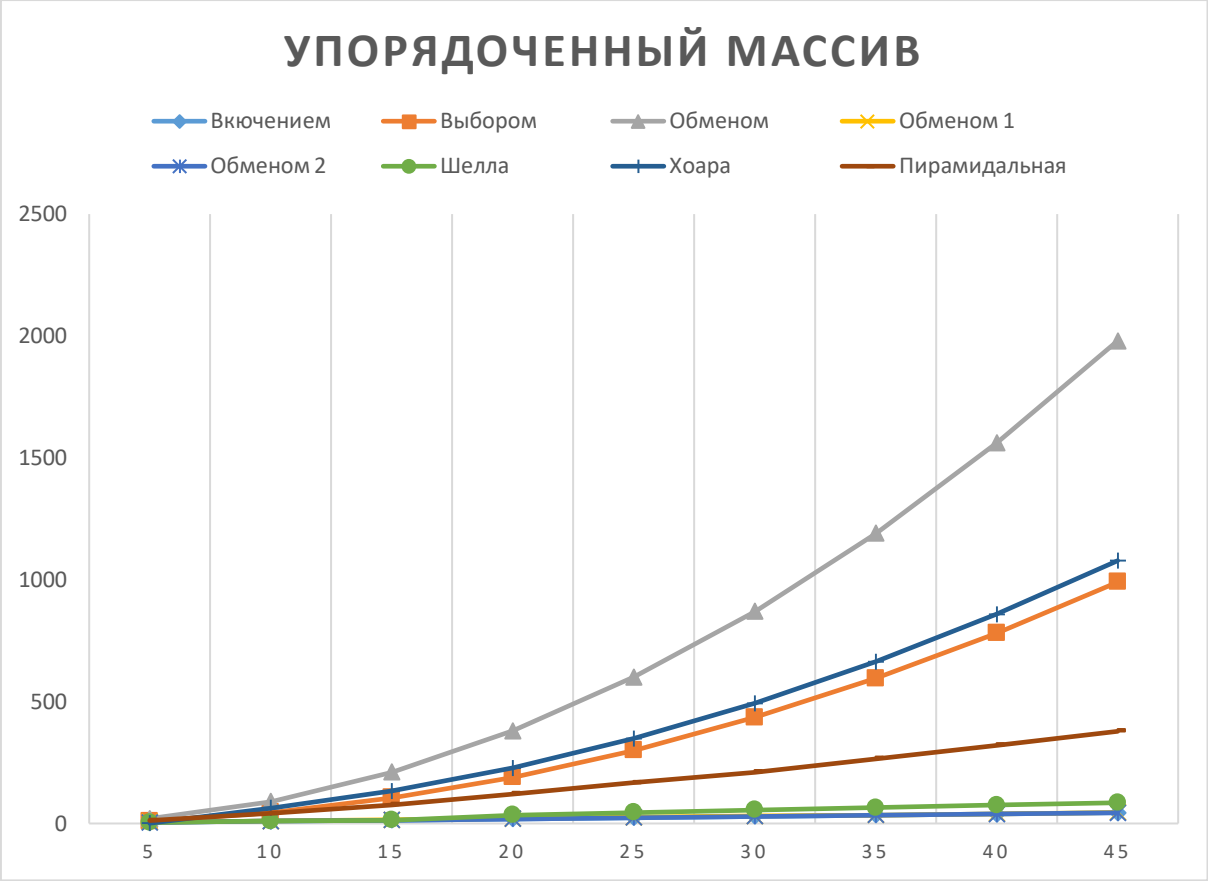
#### Неупорядоченные массивы:

Сортировка	Количество элементов в массиве								
	5	10	15	20	25	30	35	40	45
Включением	7	25	64	120	156	227	295	354	444
Выбором	10	45	105	190	300	435	595	780	990
Обменом	16	81	196	361	576	841	1156	1521	1936
Обменом 1	8	63	168	323	528	783	1088	1404	1760
Обменом 2	7	42	102	187	297	432	592	774	980
Шелла	10	32	68	89	124	156	224	275	282
Хоара	19	57	90	148	187	240	271	315	377
Пирамидальная	11	41	75	118	161	200	257	309	363

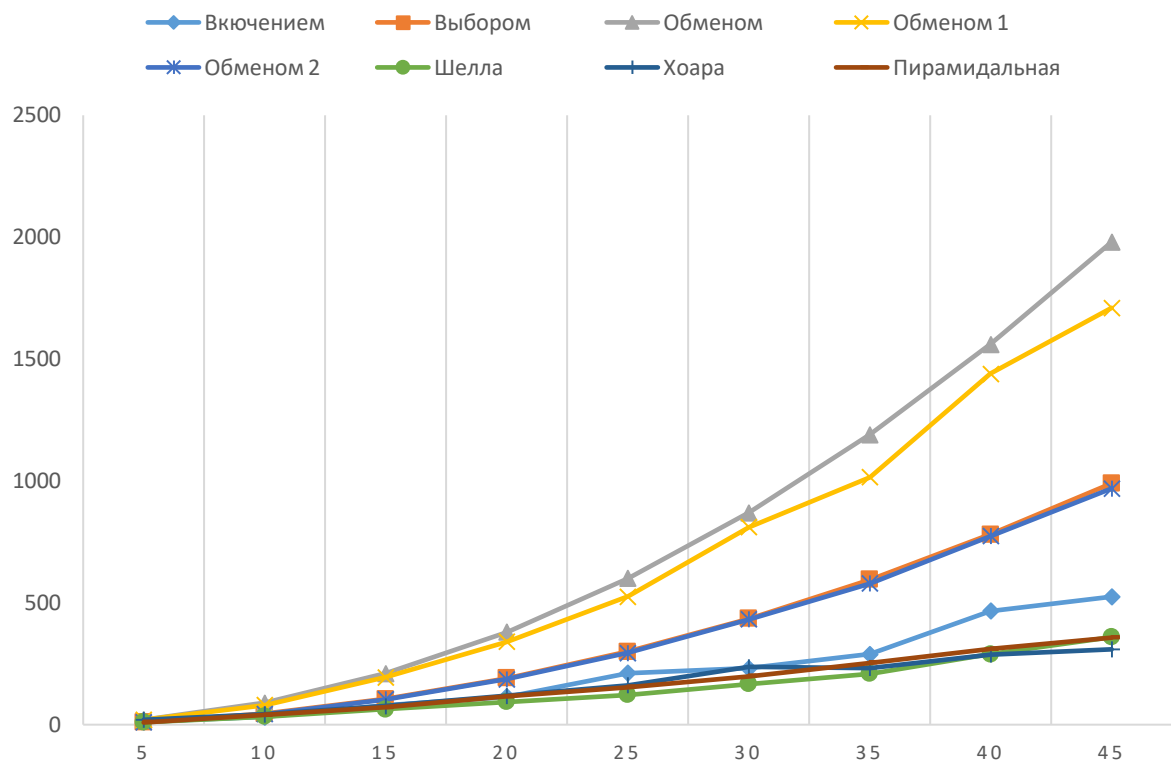
**Упорядоченные в обратном порядке массивы:**

Сортировка	Количество элементов в массиве								
	5	10	15	20	25	30	35	40	45
Включением	14	54	119	209	324	464	629	819	1034
Выбором	10	45	105	190	300	435	595	780	990
Обменом	16	81	196	361	576	841	1156	1521	1936
Обменом 1	16	81	196	361	576	841	1156	1521	1936
Обменом 2	10	45	105	190	300	435	595	780	990
Шелла	11	35	61	98	118	163	226	254	280
Хоара	20	66	140	236	360	506	680	876	1100
Пирамидальная	10	35	66	105	143	188	233	278	330

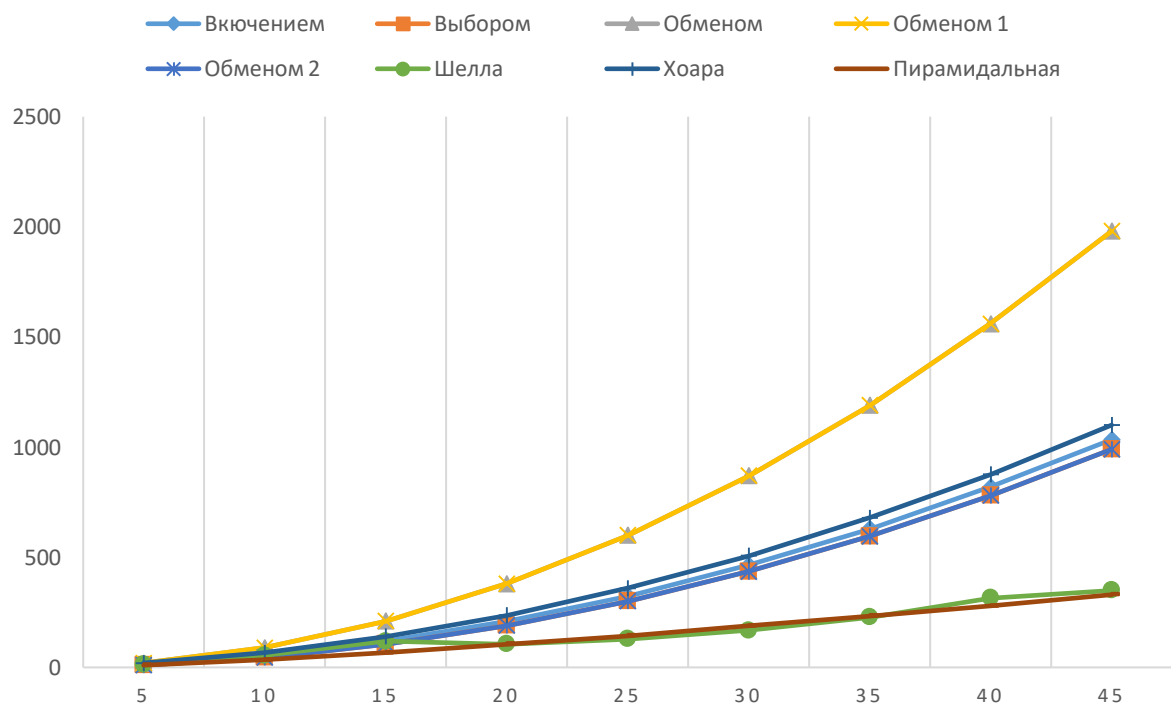
Задание 6:



## НЕУПОРЯДОЧЕННЫЙ МАССИВ



## УПОРЯДОЧЕННЫЙ В ОБРАТНОМ НАПРАВЛЕНИИ МАССИВ



### Задание 7:

- Включением – лучший случай:  $N - 1$ ; средний, худший:  $(N - 1) * N/2$
- Выбором –  $(N - 1) * N/2$
- Обменом –  $(N - 1) * N/2$
- Обменом 1 – лучший случай:  $N-1$ ; средний, худший:  $(N - 1) * N/2$
- Обменом 2 - лучший случай:  $N-1$ ; средний, худший:  $(N - 1) * N/2$
- Шелла – зависит от выбранных шагов
- Хоара –  $N + 2 * (N/2) + \dots + m * (N * m)$ ,  $m = \log_2 N$
- Пирамидальная -  $\lceil \log_2 N \rceil + \lceil \log_2 (N - 1) \rceil + \dots + \lceil \log_2 2 \rceil$

### Задание 8:

- Включением – лучший случай:  $O(N)$ ; средний, худший:  $O(N^2)$
- Выбором –  $O(N^2)$
- Обменом –  $O(N^2)$
- Обменом 1 – лучший случай:  $O(N)$ ; средний, худший:  $O(N^2)$
- Обменом 2 – лучший случай:  $O(N)$ ; средний, худший:  $O(N^2)$
- Шелла -  $O(N(\log_2 N))$ .
- Хоара – удачный разделитель –  $O(N \cdot \log_2 N)$ , в худшем -  $O(N^2)$
- Пирамидальная -  $O(N \cdot \log_2 N)$ .

### Вывод:

В ходе выполнения работы были реализованы различные методы сортировки массивов и проведен анализ количества сравнений в каждой сортировке при различных условиях заполнения массива.