

Министерство науки и высшего образования Российской Федерации  
Федеральное государственное автономное образовательное учреждение  
высшего образования  
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт цифрового развития  
Кафедра инфокоммуникаций

**ОТЧЕТ**  
**ПО ЛАБОРАТОРНОЙ РАБОТЕ №5**  
**дисциплины «Алгоритмизация»**

Выполнила:  
Кубанова Ксения Олеговна  
2 курс, группа ИВТ-б-о-22-1,  
09.03.01 «Информатика и  
вычислительная техника», очная  
форма обучения

---

(подпись)

Руководитель практики:  
Воронкин Р. А.

---

(подпись)

Отчет защищен с оценкой \_\_\_\_\_ Дата защиты \_\_\_\_\_

Ставрополь, 2023 г.

**Тема:** изучение пузырькового метода сортировки

**Цель:** понять, почему пузырьковый метод является плохим путём исследования среднего и худшего случаев.

### Порядок выполнения работы

#### Задание 1.

Написать программу, которая будет методом пузырька сортировать элементы массива в худшем случае. Худший случай в данном контексте заключается в том, что методу нужно сортировать все элементы массива.

Для исследования была составлена следующая программа:

```
const int n = 10;
int main()
{
    clock_t now = clock();
    int a[n];
    int m = 10;
    for (int i = 0; i < n; i++)
    {
        a[i] = m;
        cout << a[i] << " ";
        m = m + 10;
    }
    cout << endl;
    int d;
    for (int i = 0; i < n; i++)
    {
        for (int j = 0; j < n; j++)
        {
            if (a[i] > a[j])
            {
                d = a[i];
                a[i] = a[j];
                a[j] = d;
            }
        }
    }
    for (int i = 0; i < n; i++)
    {
        cout << a[i] << " ";
    }
    clock_t end = clock();
    double sec = (double)(end - now) / CLOCKS_PER_SEC;
    cout << endl << sec << " sec" << endl;
}
```

Рисунок 1 – программа 1

Для более точного результата было выявлено по 30 значений секундности выполнения программы, а после – их среднее. Была составлена

таблица, в которой x – размерность массива, а y – время, а последний столбец – суммы.

Таблица 1 – значения для МНК худшего случая

x	10	20	30	40	50	60	70	80	90	100	550
y	0,0487	0,027633	0,024767	0,172433	0,24767	0,2263	0,241933	0,112767	0,101667	0,226467	1,430337
x <sup>2</sup>	100	400	900	1600	2500	3600	4900	6400	8100	10000	38500
x <sup>3</sup>	1000	8000	27000	64000	125000	216000	343000	512000	729000	1000000	3025000
x <sup>4</sup>	10000	160000	810000	2560000	6250000	12960000	24010000	40960000	65610000	1E+08	253330000
xy	0,487	0,55266	0,74301	6,89732	12,3835	13,578	16,93531	9,02136	9,15003	22,6467	92,39489
x <sup>2</sup> y	4,87	11,0532	22,2903	275,8928	619,175	814,68	1185,472	721,7088	823,5027	2264,67	6743,3145

Для понимания линейности возрастания выполнения программы, будет использоваться метод наименьших квадратов.

Далее было составлено уравнение:

$$253330000a + 3025000b + 38500c = 6743,3145$$

$$\{ 30025000a + 38500b + 550c = 92,39489$$

$$38500a + 550b + 10c = 1,430337$$

В результате которого  $a = -3,5252E-05$  и  $b = 0,005445417$ ,  $c = -0,02074266$ .

Из полученных корней составляется уравнение для графика:

$$y = -3,5252E-05x^2 + 0,005445417x - 0,02074266$$

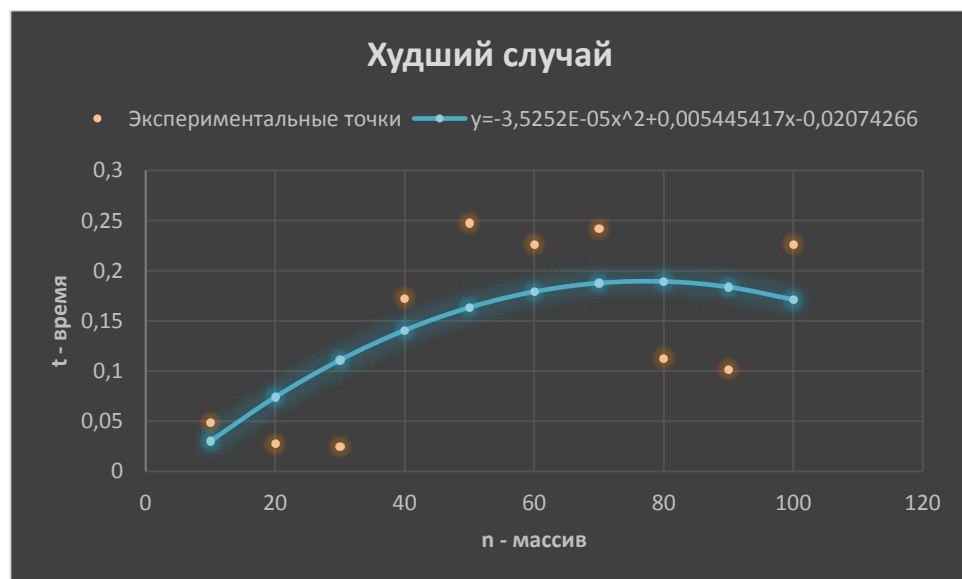


Рисунок 2 – график для худшего случая

## Задание 2.

Написать программу, которая будет методом пузырька сортировать

элементы массива в среднем случае. Средний случай в данном контексте заключается в том, что методу нужно сортировать случайные порядки элементов массива. Для исследования была составлена следующая программа:

```
const int n = 10;
int main()
{
    clock_t now = clock();
    int a[n];
    for (int i = 0; i < n; i++)
    {
        a[i] = rand() % 999 - 1;
        cout << a[i] << " ";
    }

    cout << endl;
    int d;
    for (int i = 0; i < n; i++)
    {
        for (int j = 0; j < n; j++)
        {
            if (a[i] > a[j])
            {
                d = a[i];
                a[i] = a[j];
                a[j] = d;
            }
        }
    }

    for (int i = 0; i < n; i++)
    {
        cout << a[i] << " ";
    }

    clock_t end = clock();
    double sec = (double)(end - now) / CLOCKS_PER_SEC;
    cout << endl << sec << " sec" << endl;
}
```

Рисунок 3 – программа 2

Для более точного результата было выявлено по 30 значений секундности выполнения программы, а после – их среднее. Была составлена таблица, в которой x – размерность массива, а y – время, а последний столбец – суммы.

Таблица 2 - значения для МНК среднего случая

x	10	20	30	40	50	60	70	80	90	100	550
y	0,00361 2903	0,006 097	0,010 097	0,012 806	0,017 452	0,020 097	0,024 194	0,026 452	0,029 516	0,0354 839	0,1858 0645
x 2	100	400	900	1600	2500	3600	4900	6400	8100	10000	38500
x 3	1000	8000	2700 0	6400 0	1250 00	21600 0	34300 0	51200 0	72900 0	10000 00	302500 0
x 4	10000	1600 00	8100 00	2560 000	6250 000	12960 000	24010 000	40960 000	65610 000	10000 0000	253330 000
xy	0,03612 903	0,121 935	0,302 903	0,512 258	0,872 581	1,205 806	1,693 548	2,116 129	2,656 452	3,5483 871	13,066 129

x											
2	0,36129	2,438	9,087	20,49	43,62	72,34	118,5	169,2	239,0	354,83	1030,1
y	03	71	097	032	903	839	484	903	806	871	129

Для понимания возрастания выполнения программы, будет использоваться метод наименьших квадратов.

Далее было составлено уравнение:

$$\begin{cases} 253330000a + 3025000b + 38500c = 1030,1129 \\ 3025000a + 38500b + 550c = 13,066129 \\ 38500a + 550b + 10c = 0,18580645 \end{cases}$$

В результате которого  $a = 2,67413E-06$ ,  $b = 1,30101E-05$  и  $c = 0,008138148$ . Из полученных корней составляется уравнение для графика:

$$y = 2,67413E-06x^2 + 1,30101E-05x + 0,008138148$$

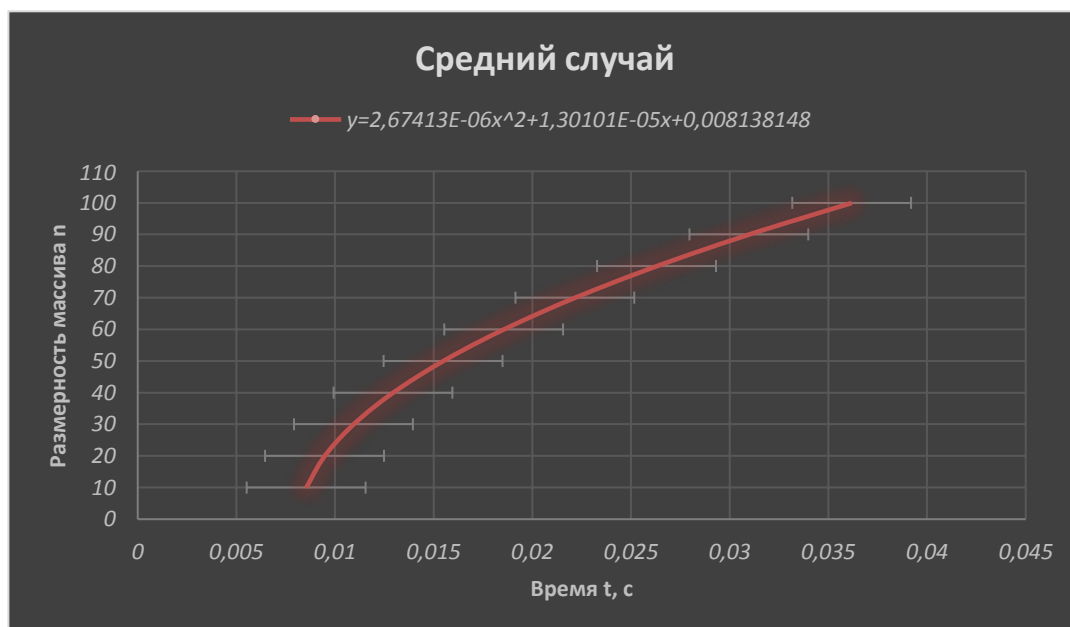


Рисунок 4 – график для среднего случая

**Вывод:** в ходе выполнения лабораторной работы было выявлено, почему пузырьковый метод сортировки является неэффективным. Во-первых, он слишком медленный. Итерации по массиву и поэтапное перемещение элементов делает сортировку медленной, что неудобно для больших размерностей массива. Во-вторых, этот метод имеет высокую сложность  $O(n^2)$ ,  $n$  – кол-во элементов), т. е. время работы алгоритма растёт быстро с увеличением массива.