

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 №9
дисциплины «Алгоритмизация»
Вариант ____

Выполнил:
Иващенко Олег Андреевич
2 курс, группа ИВТ-б-о-22-1,
09.03.02 «Информационные и
вычислительные машины»,
направленность (профиль)
«Программное обеспечение средств
вычислительной техники и
автоматизированных систем»

(подпись)

Руководитель практики:
Доцент кафедры инфокоммуникации
Воронкин Роман Александрович

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2023 г.

Тема: «Бинарный поиск»

Порядок выполнения работы

Таблица 1 – Код программы (самописные методы)

```
using System;
using System.Diagnostics;

class HelloWorld
{
    public static SearchNumber SN = new SearchNumber();

    static void Main()
    {
        // Пользовательский ввод
        Console.WriteLine("[System] Введите количество элементов массива");
        Console.Write(">>> ");
        int N = int.Parse(Console.ReadLine());
        Console.WriteLine("[System] Введите искомое число");
        Console.Write(">>> ");
        int searchNumber = int.Parse(Console.ReadLine());

        // Генерация значений в массив и сортировка
        Stopwatch Timer = Stopwatch.StartNew();
        Timer.Start();
        int[] array = new int[N];
        Random rnd = new Random();
        for (int i = 0; i < array.Length; i++) array[i] = rnd.Next(-500, 500);
        array = ArraySort(array);

        // Вывод элементов, если их количество менее 30
        if (N < 30) for (int i = 0; i < array.Length; i++) Console.WriteLine($"[{i}] {array[i]}");

        SN = Binar(array, searchNumber, 0, array.Length - 1);
        if (SN.ID == -1) Console.WriteLine("[System] Искомое значения не было найдено");
        else Console.WriteLine($"[System] Искомое значение: [{SN.ID}] {SN.Number}");
        Console.WriteLine($"[System] Время выполнения: {Timer.Elapsed.TotalSeconds} сек.");
        Timer.Stop();
        Console.ReadKey();
    }

    // Метод сортировки
    static int[] ArraySort(int[] array)
    {
        for (int i = 0; i < array.Length; i++)
            for (int j = 0; j < array.Length; j++)
                if (array[j] > array[i])
                {
                    int temp = array[i];
                    array[i] = array[j];
                    array[j] = temp;
                }
        return array;
    }

    // Метод бинарного поиска
    static SearchNumber Binar(int[] array, int number, int left, int right)
    {
        SearchNumber temp = new SearchNumber();
```

```

while (left <= right)
{
    int mid = left + (right - left) / 2;

    if (array[mid] == number)
    {
        temp = new SearchNumber(mid, number);
        break;
    }

    if (array[mid] < number) left = mid + 1;
    else right = mid - 1;
}
return temp;
}
}

/*
Класс, созданный для упрощения работы со значениями.
Экземпляр содержит значения ID и Number, которые являются числовым
идентификатором и значением экземпляра соответственно.
*/
class SearchNumber
{
    public int ID;
    public int Number;

    public SearchNumber()
    {
        ID = -1;
        Number = int.MinValue;
    }

    public SearchNumber(int ID, int Number)
    {
        this.ID = ID;
        this.Number = Number;
    }
}

```

Таблица 2 – Код программы (встроенные функции)

```

using System;
using System.Diagnostics;

class HelloWorld
{
    static void Main()
    {
        Console.WriteLine("[System] Введите количество элементов массива");
        Console.Write(">>> ");
        int N = int.Parse(Console.ReadLine()); Console.WriteLine("[System] Введите искомое число");
        Console.Write(">>> ");
        int searchNumber = int.Parse(Console.ReadLine());

        Stopwatch Timer = Stopwatch.StartNew();
        Timer.Start();
        int[] array = new int[N];
        Random rnd = new Random();
        for (int i = 0; i < array.Length; i++) array[i] = rnd.Next(-500, 500);
    }
}

```

```

Array.Sort(array); // Сортировка

if (N < 30)
    for (int i = 0; i < array.Length; i++) Console.WriteLine($"[{i}] {array[i]}");

int searchID = Array.BinarySearch(array, searchNumber);
if (searchID > 0) Console.WriteLine($"[System] Искомое значение: [{Array.BinarySearch(array,
searchNumber)}] {searchNumber}");
else Console.WriteLine($"[System] Искомое значение не найдено");
Console.WriteLine($"[System] Время выполнения: {Timer.Elapsed.TotalSeconds}");
Timer.Stop();
Console.ReadKey();
}
}

```

```

[System] Введите количество элементов массива
>>> 20
[System] Введите искомое число
>>> 0
[0] -46
[1] -41
[2] -34
[3] -23
[4] -15
[5] -9
[6] -4
[7] 0
[8] 2
[9] 3
[10] 15
[11] 18
[12] 21
[13] 22
[14] 25
[15] 27
[16] 27
[17] 28
[18] 30
[19] 48
[System] Искомое значение: [7] 0
[System] Время выполнения: 0,5128127 сек.

```

Рисунок 1 – Результат выполнения программы

Самописные функции сортировки и бинарного поиска											Сумма
Размерность массива M	100	200	300	400	500	600	700	800	900	1000	5500
Время выполнения T, сек	0,0521425	0,006163	0,063014	0,087268	0,004394	0,003434	0,007942	0,016143	0,014528	0,01809	0,273118
M * M	10000	40000	90000	160000	250000	360000	490000	640000	810000	1000000	3850000
M * T	5,21425	1,23258	18,9042	34,90724	2,1968	2,0604	5,55947	12,91432	13,07484	18,0902	114,1543
γ	-0,00017083	-0,00021	-0,00024	-0,00026	-0,00027	-0,00028	-0,00029	-0,0003	-0,00031	-0,00032	-0,00266
Корреляция	0.135273527										

Рисунок 2 – Таблица значений для самописных функций сортировки массива и бинарного поиска значений для средних случаев

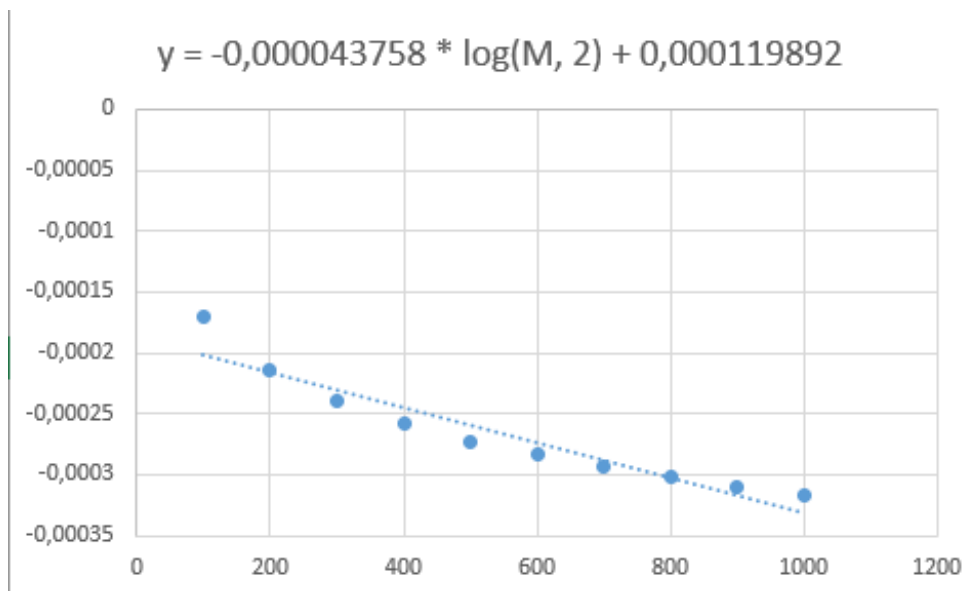


Рисунок 3 – График функции $y = -0,000043758 * \log(M, 2) + 0,000119892$ для средних исходов программы (значение найдено в середине массива)

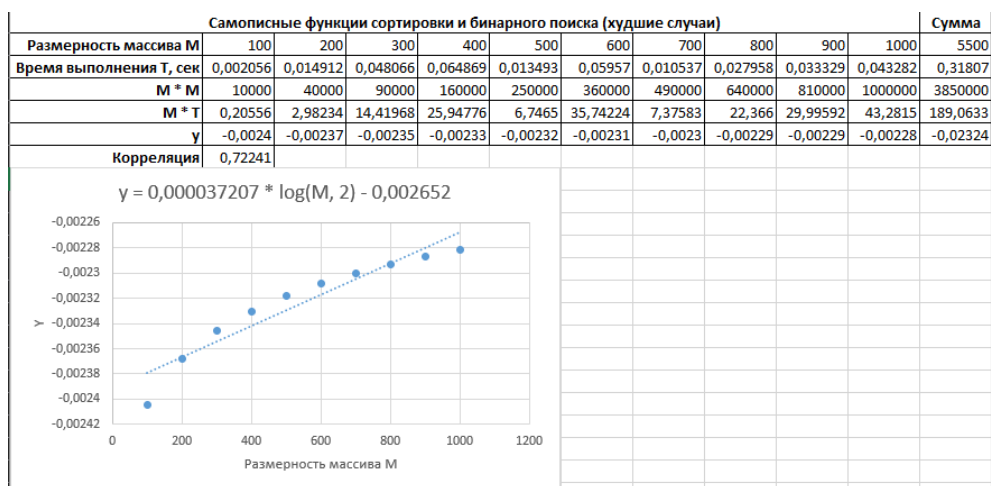


Рисунок 4 – Таблица значений и график функции для худших случаев (искомое значение последнее в массиве) самописных методов

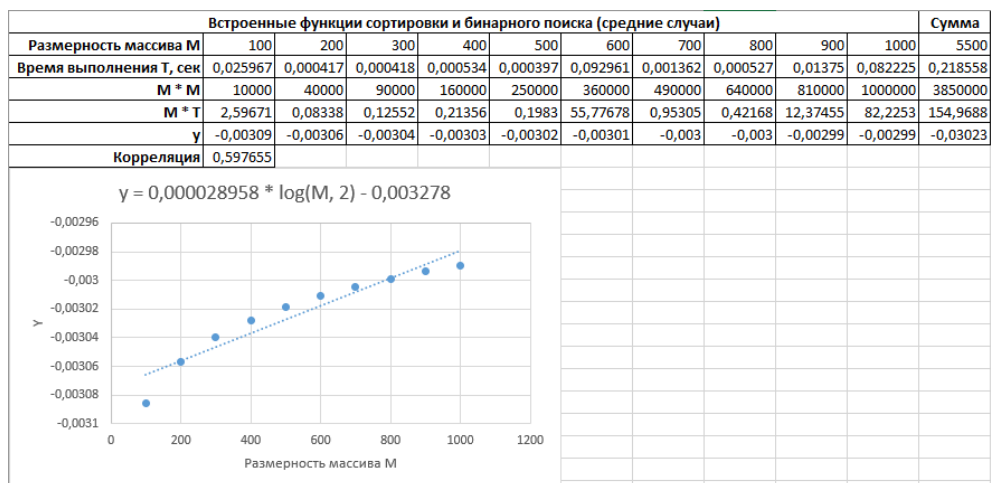


Рисунок 5 – Таблица значений и график функции для средних случаев
встроенных методов

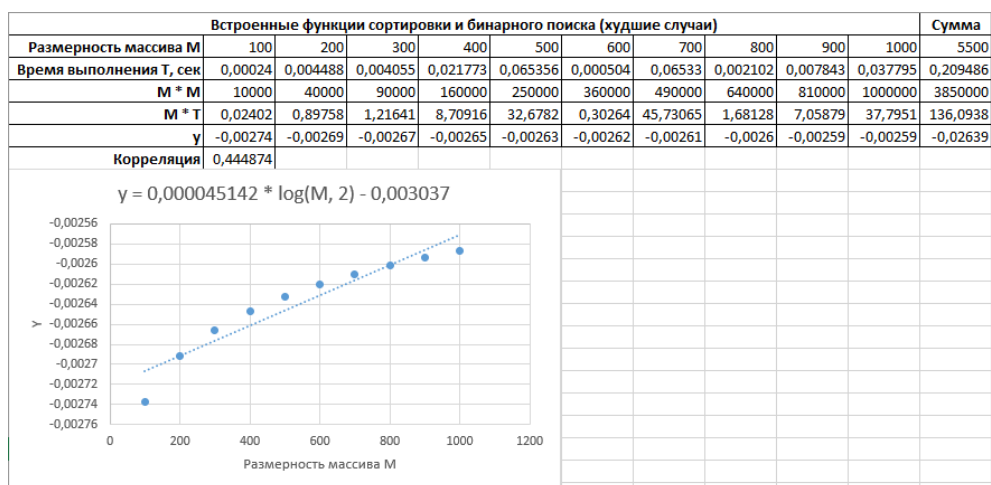


Рисунок 6 – Таблица значений и график функции для худших случаев
встроенных методов