

**Министерство образования и науки Российской Федерации**  
**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ**  
**УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ,**  
**МЕХАНИКИ И ОПТИКИ**

Факультет программной инженерии и компьютерной техники  
Направление подготовки 09.03.04 Программная инженерия

Дисциплина «Алгоритмы и структуры данных»

**ОТЧЁТ**

по лабораторной работе №3

Студент Луговских Савелий Р3218

Преподаватель Муромцев Дмитрий Ильич

Санкт-Петербург

2019 г.

# Задача на программирование: двоичный поиск

В первой строке даны целое число  $1 \leq n \leq 10^5$  и массив  $A[1 \dots n]$  из  $n$  различных натуральных чисел, не превышающих  $10^9$ , в порядке возрастания, во второй — целое число  $1 \leq k \leq 10^5$  и  $k$  натуральных чисел  $b_1, \dots, b_k$ , не превышающих  $10^9$ . Для каждого  $i$  от 1 до  $k$  необходимо вывести индекс  $1 \leq j \leq n$ , для которого  $A[j] = b_i$ , или  $-1$ , если такого  $j$  нет.

```
using System;  
using System.Collections.Generic;  
using System.IO;
```

```
namespace Stepik  
{  
    public class BinarySearch  
    {  
        private int[] array;  
  
        public BinarySearch(int[] array)  
        {  
            this.array = array;  
        }  
  
        public int Find(int value)  
        {  
            int left = -1;  
            int right = array.Length - 1;  
            while (left != right - 1)  
            {  
                int middle = (right + left) / 2;  
                if (value > array[middle])  
                    left = middle;  
                else  
                    right = middle;  
            }  
            if (array[right] != value)  
                return -1;  
            return right + 1;  
        }  
    }  
  
    public class BinarySearchTask  
    {  
        static void Main(string[] args)  
        {  
            string[] str = Console.ReadLine().Split(' ');  
            int n = int.Parse(str[0]);  
            int[] array = new int[n];  
            for (int i = 0; i < n; i++)  
                array[i] = int.Parse(str[i + 1]);  
  
            BinarySearch BinarySearch = new BinarySearch(array);  
            str = Console.ReadLine().Split(' ');  
            n = int.Parse(str[0]);  
            array = new int[n];
```

```

        for (int i = 0; i < n; i++)
            array[i] = BinarySearch.Find(int.Parse(str[i + 1]));

        for (int i = 0; i < n; i++)
            Console.Write("{0} ", array[i]);
    }
}
}

```

## Задача на программирование: число инверсий

Первая строка содержит число  $1 \leq n \leq 10^5$ , вторая — массив  $A[1 \dots n]$ , содержащий натуральные числа, не превосходящие  $10^9$ . Необходимо посчитать число пар индексов  $1 \leq i < j \leq n$ , для которых  $A[i] > A[j]$ . (Такая пара элементов называется инверсией массива. Количество инверсий в массиве является в некотором смысле его мерой неупорядоченности: например, в упорядоченном по неубыванию массиве инверсий нет вообще, а в массиве, упорядоченном по убыванию, инверсию образуют каждые два элемента.)

```

using System;
using System.Collections.Generic;
using System.IO;

namespace Stepik
{
    public class InversionsAmount
    {
        private int[] array;

        public InversionsAmount(int[] array)
        {
            this.array = array;
        }

        private long MergeArrays(int[] bufferArray, int firstArrayBegin, int firstArrayEnd, int secondArrayBegin, int secondArrayEnd)
        {
            int sortedAmount = 0;
            int begin = firstArrayBegin;

            long inversionsAmount = 0;

            while (firstArrayBegin <= firstArrayEnd && secondArrayBegin <= secondArrayEnd)
            {
                if (array[firstArrayBegin] <= array[secondArrayBegin])
                {
                    bufferArray[sortedAmount] = array[firstArrayBegin];
                    firstArrayBegin++;
                }
                else
                {
                    bufferArray[sortedAmount] = array[secondArrayBegin];
                    secondArrayBegin++;
                    inversionsAmount += firstArrayEnd - firstArrayBegin + 1;
                }
                sortedAmount++;
            }
            for (int i = firstArrayBegin; i <= firstArrayEnd; i++)
            {
                bufferArray[sortedAmount] = array[i];
            }
        }
    }
}

```

```

        sortedAmount++;
    }
    for (int i = secondArrayBegin; i <= secondArrayEnd; i++)
    {
        bufferArray[sortedAmount] = array[i];
        sortedAmount++;
    }

    Array.Copy(bufferArray, 0, array, begin, sortedAmount);
    return inversionsAmount;
}

public long Sort()
{
    int[] bufferArray = new int[array.Length];
    long inversionsAmount = 0;
    int sortSize = 2;

    while (sortSize < array.Length * 2)
    {
        int firstArrayBegin = 0;
        while (firstArrayBegin < array.Length)
        {
            int secondArrayEnd = firstArrayBegin + sortSize - 1;

            int firstArrayEnd = (firstArrayBegin + secondArrayEnd) / 2 - (firstArrayBegin + secondArrayEnd + 1)
% 2;
            int secondArrayBegin = firstArrayEnd + 1;

            if (secondArrayBegin < array.Length)
            {
                if (secondArrayEnd >= array.Length)
                    secondArrayEnd = array.Length - 1;
                inversionsAmount += MergeArrays(bufferArray, firstArrayBegin, firstArrayEnd, secondArrayBegin,
secondArrayEnd);
            }
            firstArrayBegin += sortSize;
        }
        sortSize *= 2;
    }
    return inversionsAmount;
}

}

public class InversionsAmountTask
{
    static void Main(string[] args)
    {
        int n = int.Parse(Console.ReadLine());
        int[] array = new int[n];
        string[] str = Console.ReadLine().Split(' ');
        for (int i = 0; i < n; i++)
            array[i] = int.Parse(str[i]);

        InversionsAmount inversionsAmount = new InversionsAmount(array);
        Console.WriteLine(inversionsAmount.Sort());
    }
}
}

```

# Задача на программирование: сортировка подсчётом

Первая строка содержит число  $1 \leq n \leq 10^4$ , вторая —  $n$  натуральных чисел, не превышающих 10. Выведите упорядоченную по неубыванию последовательность этих чисел.

```
using System;
using System.Collections.Generic;
using System.IO;

namespace Stepik.DivideAndRule
{
    public class CountingSorter
    {
        private int[] array;

        public CountingSorter(int[] array)
        {
            this.array = array;
        }

        public int[] Sort(int max)
        {
            int[] buffer = new int[max + 1];

            for (int i = 0; i < array.Length; i++)
                buffer[array[i]]++;

            int currentSize = 0;
            for (int i = 0; i < buffer.Length; i++)
            {
                while (buffer[i] > 0)
                {
                    array[currentSize] = i;
                    buffer[i]--;
                    currentSize++;
                }
            }

            return array;
        }
    }

    public class CountingSortTask
    {
        static void Main(string[] args)
        {
            int n = int.Parse(Console.ReadLine());
            string[] str = Console.ReadLine().Split(' ');
            int[] array = new int[n];
            int max = 0;
            for (int i = 0; i < n; i++)
            {
                array[i] = int.Parse(str[i]);
                max = Math.Max(max, array[i]);
            }

            CountingSorter sorter = new CountingSorter(array);
            array = sorter.Sort(max);

            for (int i = 0; i < array.Length; i++)
```

```
        Console.WriteLine("{0} ", array[i]);  
    }  
}
```