

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 ПРАКТИЧЕСКОЙ РАБОТЕ №8
дисциплины «Алгоритмизация»

Выполнил:

Лейс Алексей Вячеславович
2 курс, группа ИВТ-б-о-22-1,
09.03.01 «Информатика и
вычислительная техника»,
направленность (профиль)
«Программное обеспечение средств
вычислительной техники и
автоматизированных систем»

(подпись)

Руководитель практики кандидат тех.
наук доцент кафедры
инфокоммуникаций: Воронкин Р.А

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2023 г.

Порядок выполнения работы:

Посчитать количество инверсий в массиве, то есть минимальное количество перестановок. $O(n \log n)$

```
1 public static int CountInversions(int[] arr)
2 {
3     if (arr == null || arr.Length <= 1)
4     {
5         return 0; // Нет инверсий в пустом массиве или массиве из одного элемента
6     }
7
8     int[] temp = new int[arr.Length];
9     return MergeSortAndCount(arr, temp, 0, arr.Length - 1);
10 }
11
12 private static int MergeSortAndCount(int[] arr, int[] temp, int left, int right)
13 {
14     int count = 0;
15
16     if (left < right)
17     {
18         int mid = (left + right) / 2;
19
20         // Рекурсивно сортируем и подсчитываем инверсии в левой и правой половинах массива
21         count += MergeSortAndCount(arr, temp, left, mid);
22         count += MergeSortAndCount(arr, temp, mid + 1, right);
23
24         // Объединяем две отсортированных половины и подсчитываем инверсии
25         count += MergeAndCount(arr, temp, left, mid, right);
26     }
27
28     return count;
29 }
30
31 private static int MergeAndCount(int[] arr, int[] temp, int left, int mid, int right)
32 {
33     int i = left;
34     int j = mid + 1;
35     int k = left;
36     int count = 0;
37
38     while (i <= mid && j <= right)
39     {
40         if (arr[i] <= arr[j])
41         {
42             temp[k++] = arr[i++];
43         }
44         else
45         {
46             // Если arr[i] > arr[j], то у нас есть инверсии
47             temp[k++] = arr[j++];
48             count += (mid - i + 1);
49         }
50     }
51
52     // Копируем оставшиеся элементы, если они есть
53     while (i <= mid)
54     {
55         temp[k++] = arr[i++];
56     }
57
58     while (j <= right)
59     {
60         temp[k++] = arr[j++];
61     }
62
63     // Копируем отсортированные элементы обратно в оригинальный массив
64     for (int l = left; l <= right; l++)
65     {
66         arr[l] = temp[l];
67     }
68
69     return count;
70 }
71
72 static void Main()
73 {
74     int[] arr = { 1, 20, 9, 4, 5 };
75     int inversionCount = CountInversions(arr);
76
77     Console.WriteLine("Количество инверсий в массиве: " + inversionCount);
78 }
79
```

```
42 while (i <= mid && j <= right)
43 {
44     if (arr[i] <= arr[j])
45     {
46         temp[k++] = arr[i++];
47     }
48     else
49     {
50         // Если arr[i] > arr[j], то у нас есть инверсии
51         temp[k++] = arr[j++];
52         count += (mid - i + 1);
53     }
54 }
55
56 // Копируем оставшиеся элементы, если они есть
57 while (i <= mid)
58 {
59     temp[k++] = arr[i++];
60 }
61
62 while (j <= right)
63 {
64     temp[k++] = arr[j++];
65 }
66
67 // Копируем отсортированные элементы обратно в оригинальный массив
68 for (int l = left; l <= right; l++)
69 {
70     arr[l] = temp[l];
71 }
72
73 return count;
74 }
75
76 static void Main()
77 {
78     int[] arr = { 1, 20, 9, 4, 5 };
79     int inversionCount = CountInversions(arr);
80
81     Console.WriteLine("Количество инверсий в массиве: " + inversionCount);
82 }
83
```

Количество инверсий в массиве: 5