

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»**  
**Вариант \_\_\_\_**

Выполнил:  
Репкин Александр Павлович  
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», очная форма обучения

---

(подпись)

Руководитель практики:  
Воронкин Р.А., канд. техн. наук,  
доцент кафедры инфокоммуникаций

---

(подпись)

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

Ставрополь, 2023 г.

## Порядок выполнения работы:

1. Создана программа на основе поставленного задания — необходимо посчитать количество инверсий в массиве (минимальное количество перестановок, после выполнения которых массив будет в отсортированном виде), при этом время выполнения не должно превышать  $O(n \cdot \log(n))$ . Для решения поставленной задачи выбран язык программирования Java. В коде использован метод сортировки слиянием, на который требуется  $O(n \cdot \log(n))$  времени. Причина, по которой используется метод сортировки слиянием, заключается в том, что при слиянии двух упорядоченных массивов (leftArray и rightArray), элементы в левой части считаются уже упорядоченными, и если элемент из правой части меньше, то элемент из правой части образует инверсию с оставшимися элементами в левой части.

```
/* Сортировка слиянием, время работы =  $O(n \cdot \log(n))$ .  
Тут происходит разделение массива на две части (от start до middle и от middle до finish), которые  
сортируются по отдельности, после чего их объединяет функция merge.  
Повторяется, пока весь массив не будет отсортирован.  
*/  
static void mergeSort(ArrayList<Integer> array, int start, int finish) { 3 usages  
    if (start < finish) {  
        int middle = (start + finish) / 2;  
        mergeSort(array, start, middle);  
        mergeSort(array, start: middle + 1, finish);  
        merge(array, start, middle, finish);  
    }  
}
```

Рисунок 1. Полученный код сортировки слиянием.

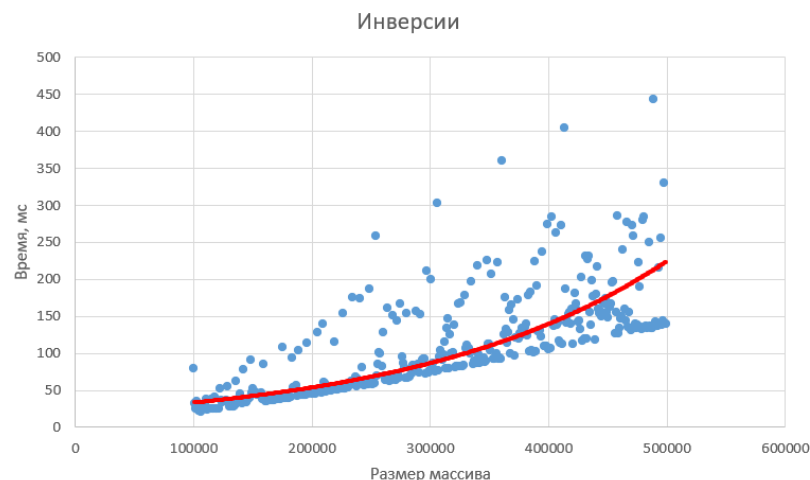


Рисунок 2. Полученный график счёта количества инверсий.

**Вывод:** в ходе выполнения практической работы был рассмотрен более быстрый способ подсчёта инверсий в массиве. Полученный метод гораздо быстрее стандартного подсчёта инверсий, включающего в себя перебор массива при помощи двух циклов for, ведь такой способ затрачивает  $O(n) * O(n) = O(n^2)$  времени.