

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 8**

Выполнил:  
Данилецкий Дмитрий Витальевич  
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. технических  
наук, доцент кафедры  
инфокоммуникаций

---

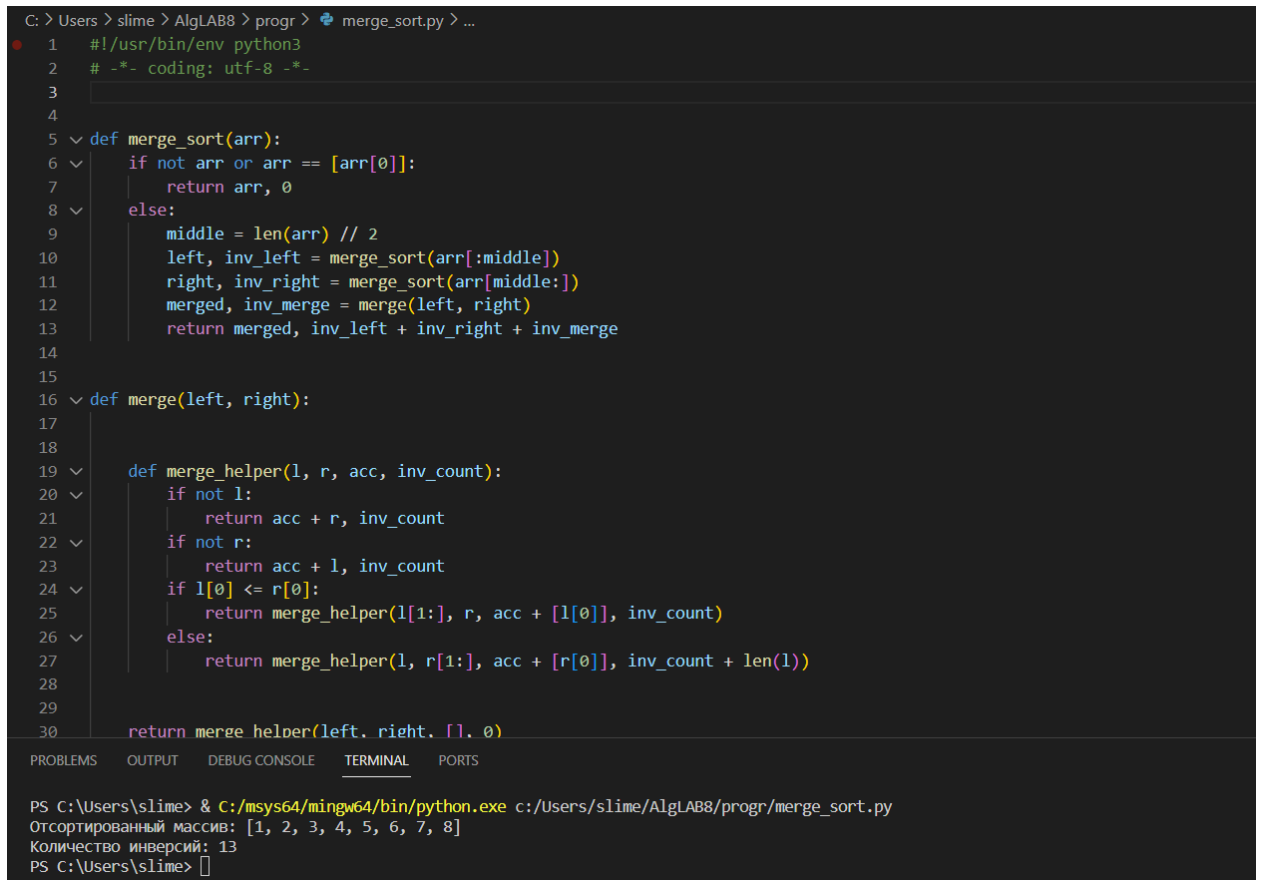
(подпись)

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

Ставрополь, 2023 г.

## Ход работы

1. Написал программу для подсчета инверсий в массиве чисел за время выполнения  $O(n \cdot \log(n))$



```
C: > Users > slime > AlgLAB8 > progr > merge_sort.py > ...
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4
5  def merge_sort(arr):
6      if not arr or arr == [arr[0]]:
7          return arr, 0
8      else:
9          middle = len(arr) // 2
10         left, inv_left = merge_sort(arr[:middle])
11         right, inv_right = merge_sort(arr[middle:])
12         merged, inv_merge = merge(left, right)
13         return merged, inv_left + inv_right + inv_merge
14
15
16  def merge(left, right):
17
18      def merge_helper(l, r, acc, inv_count):
19          if not l:
20              return acc + r, inv_count
21          if not r:
22              return acc + l, inv_count
23          if l[0] <= r[0]:
24              return merge_helper(l[1:], r, acc + [l[0]], inv_count)
25          else:
26              return merge_helper(l, r[1:], acc + [r[0]], inv_count + len(l))
27
28      return merge_helper(left, right, [], 0)
29
30
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
PS C:\Users\slime> & c:/msys64/mingw64/bin/python.exe c:/Users/slime/AlgLAB8/progr/merge_sort.py
Отсортированный массив: [1, 2, 3, 4, 5, 6, 7, 8]
Количество инверсий: 13
PS C:\Users\slime>
```

Рисунок 1. Результат выполнения программы

Вывод: в ходе выполнения лабораторной работы было исследовано вычисление числа инверсий в массиве. Оказалось, что интуитивный метод, основанный на простом переборе, имеет квадратичную временную сложность, что может быть неэффективным для больших массивов. В результате исследований был выбран более эффективный подход — использование метода сортировки слиянием с подсчетом инверсий. Этот метод демонстрирует временную сложность  $O(n \cdot \log(n))$ , что позволяет значительно ускорить вычисления и обрабатывать массивы большего размера более эффективно. Таким образом, использование сортировки слиянием с подсчетом инверсий является более оптимальным решением для задачи подсчета инверсий в массиве.