

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 №8
дисциплины «Алгоритмизация»

Выполнил:
Зармухамбетов Булат
Эльдарович
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., доцент кафедры
инфокоммуникаций

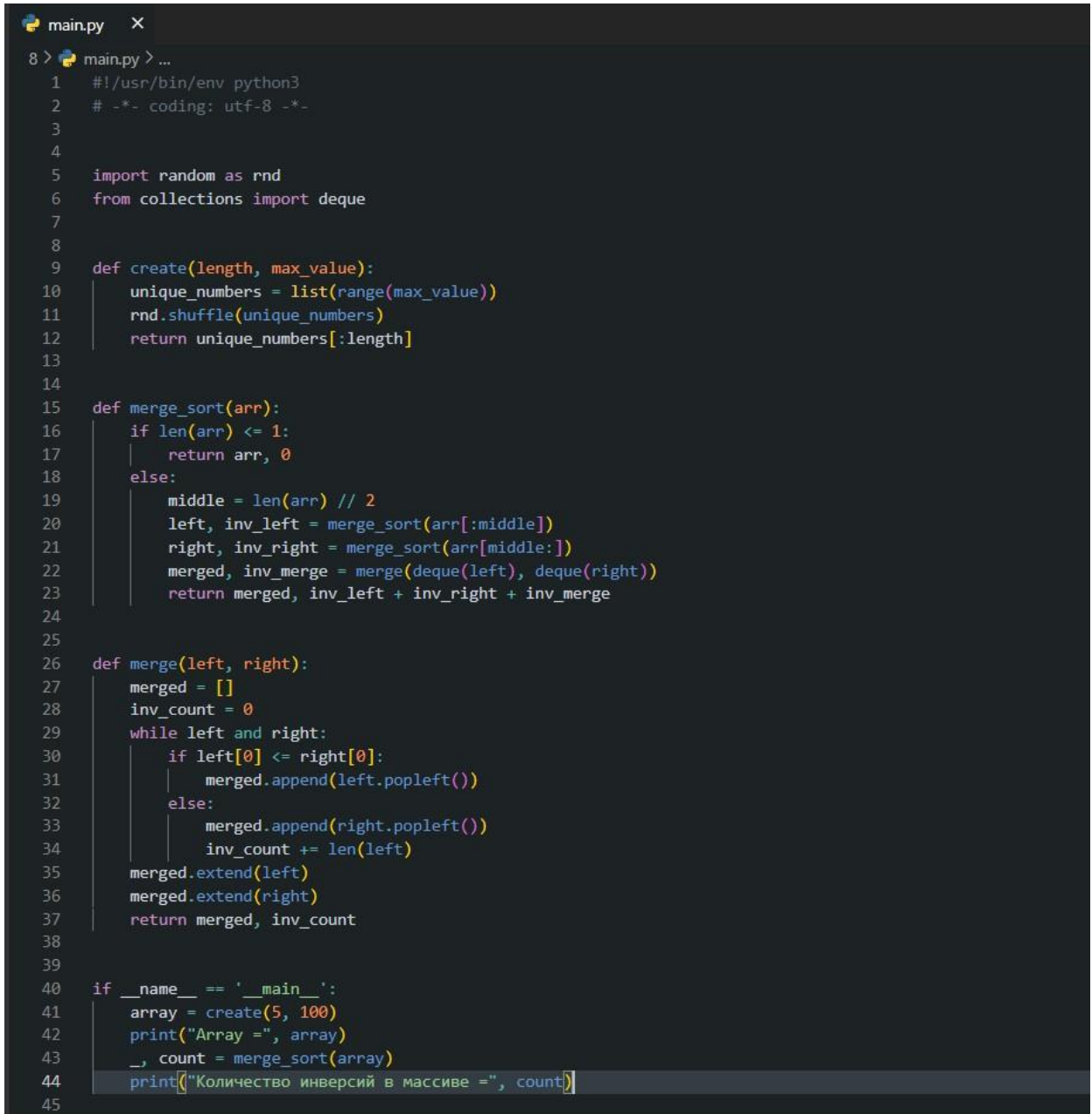
(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2023 г.

Порядок выполнения работы:

1. Написал программу подсчета инверсий в программе за время $O(n \cdot \log(n))$:



```
main.py X
8 > main.py > ...
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4
5  import random as rnd
6  from collections import deque
7
8
9  def create(length, max_value):
10     unique_numbers = list(range(max_value))
11     rnd.shuffle(unique_numbers)
12     return unique_numbers[:length]
13
14
15  def merge_sort(arr):
16     if len(arr) <= 1:
17         return arr, 0
18     else:
19         middle = len(arr) // 2
20         left, inv_left = merge_sort(arr[:middle])
21         right, inv_right = merge_sort(arr[middle:])
22         merged, inv_merge = merge(deque(left), deque(right))
23         return merged, inv_left + inv_right + inv_merge
24
25
26  def merge(left, right):
27     merged = []
28     inv_count = 0
29     while left and right:
30         if left[0] <= right[0]:
31             merged.append(left.popleft())
32         else:
33             merged.append(right.popleft())
34             inv_count += len(left)
35     merged.extend(left)
36     merged.extend(right)
37     return merged, inv_count
38
39
40  if __name__ == '__main__':
41     array = create(5, 100)
42     print("Array =", array)
43     _, count = merge_sort(array)
44     print("Количество инверсий в массиве =", count)
45
```

Рисунок 1. Код программы

```
PS C:\Users\dalam\OneDrive\Рабочий стол\projects\projects\Алгоритмизация> python
Array = [86, 42, 64, 60, 65]
Количество инверсий в массиве = 5
PS C:\Users\dalam\OneDrive\Рабочий стол\projects\projects\Алгоритмизация> python
Array = [49, 46, 65, 20, 4]
Количество инверсий в массиве = 8
PS C:\Users\dalam\OneDrive\Рабочий стол\projects\projects\Алгоритмизация> python
Array = [3, 4, 28, 68, 36]
Количество инверсий в массиве = 1
PS C:\Users\dalam\OneDrive\Рабочий стол\projects\projects\Алгоритмизация> python
Array = [89, 12, 23, 96, 74]
Количество инверсий в массиве = 4
PS C:\Users\dalam\OneDrive\Рабочий стол\projects\projects\Алгоритмизация> python
Array = [60, 45, 99, 81, 85]
Количество инверсий в массиве = 3
PS C:\Users\dalam\OneDrive\Рабочий стол\projects\projects\Алгоритмизация>
```

Рисунок 2. Результат работы программы

Вывод: в результате выполнения лабораторной работы был изучен алгоритм mergesort и способ встроения в него подсчета инверсий.