

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:  
Кожуховский Виктор Андреевич  
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 Роман Александрович

---

(подпись)

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

Ставрополь, 2023 г.

Порядок выполнения работы:

Посчитать количество инверсий в массиве, перестановок после выполнения которых массив будет отсортирован, время выполнения не должно превышать  $O(n) \cdot \log(n)$ .

```
1 #!/usr/bin/env python3
2 # coding: utf-8 -*-
3
4 import random
5
6
7 def merge_sort(alist):
8     if len(alist) > 1:
9         mid = len(alist)//2 # Разделение пополам
10        left_half = alist[:mid]
11        right_half = alist[mid:]
12
13        # Сортировка половин и подсчет инверсий
14        left_half, left_inversions = merge_sort(left_half)
15        right_half, right_inversions = merge_sort(right_half)
16        total_inversions = left_inversions + right_inversions
17
18        i, j, k = 0, 0, 0
19        while i < len(left_half) and j < len(right_half):
20            if left_half[i] <= right_half[j]:
21                alist[k] = left_half[i]
22                i += 1
23            else:
24                alist[k] = right_half[j]
25                j += 1
26                total_inversions += len(left_half) - i
27                k += 1
28
29        while i < len(left_half):
30            alist[k] = left_half[i]
31            i += 1
32            k += 1
33
34        while j < len(right_half):
35            alist[k] = right_half[j]
36            j += 1
37            k += 1
38
39        return alist, total_inversions
40
41    return alist, 0
42
43
44 def fill_list(num_of_elements):
45     a = [random.randint(0, 1000) for _ in range(num_of_elements)]
46     return a
47
48
49 if __name__ == "__main__":
50     a = fill_list(100)
51     print(a)
52     a, cnt = merge_sort(a)
53     print("Reps: ", cnt)
54     print(a)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS Python + - [ ] ... ^ x

PS C:\Users\viktor> & "C:/Program Files/Python311/python.exe" c:/Users/viktor/Desktop/схемы/алгоритмизация/AlgLab8/Main.py

[127, 343, 769, 34, 718, 993, 2, 959, 561, 510, 645, 817, 314, 961, 442, 772, 363, 217, 273, 748, 168, 581, 770, 211, 562, 190, 604, 320, 738, 580, 262, 460, 948, 229, 673, 129, 866, 218, 313, 606, 3, 767, 978, 225, 833, 231, 228, 218, 734, 626, 615, 874, 140, 698, 83, 730, 789, 805, 653, 348, 908, 190, 464, 332, 236, 421, 858, 512, 267, 750, 2, 463, 69, 75, 881, 849, 421, 671, 912, 737, 378, 846, 990, 410, 308, 131, 616, 979, 439, 186, 422, 589, 693, 295, 190, 646, 131, 522, 927, 481]

Reps: 2447

[2, 2, 3, 34, 69, 75, 83, 127, 129, 131, 131, 140, 168, 186, 190, 190, 190, 211, 217, 218, 218, 225, 228, 229, 231, 236, 262, 267, 273, 295, 308, 313, 314, 320, 332, 343, 348, 363, 378, 410, 421, 421, 422, 439, 442, 460, 463, 464, 481, 510, 512, 522, 561, 562, 580, 581, 589, 604, 606, 615, 616, 626, 645, 646, 653, 671, 673, 693, 698, 718, 730, 734, 737, 738, 748, 750, 767, 769, 770, 772, 789, 805, 817, 833, 846, 849, 858, 866, 874, 881, 908, 912, 927, 948, 959, 961, 978, 979, 990, 993]

Рисунок 1. Результат выполнения и код программы сортировки при помощи алгоритма mergesort со встроенным в него подсчетом инверсий

Написал программу сортировки при помощи алгоритма mergesort со встроенным в него подсчетом инверсий и последующем выводе неотсортированного массива, количества инверсий и отсортированного массива.

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