

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_arr(numOfEl):
45     a = [random.randint(0, 1000) for _ in range(numOfEl)]
46     return a
47
48
49 if __name__ == "__main__":
50     a = fill_arr(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 + - [icon] [icon] [icon] [icon] [icon]

PS C:\Users\viktor> & "C:/Program Files/Python311/python.exe" c:/Users/viktor/Desktop/скфп/алгоритмизация/AlgLab8/Main.py
[800, 643, 418, 382, 190, 170, 197, 904, 779, 812, 135, 145, 514, 30, 95, 170, 908, 293, 334, 119, 260, 227, 846, 702, 476, 368, 40, 330, 48, 889, 274, 551, 678, 237, 120, 151, 39, 6, 910, 498, 610, 660, 133, 409, 361, 489, 495, 684, 634, 926, 566, 114, 242, 141, 929, 273, 250, 775, 713, 719, 805, 204, 292, 834, 209, 45, 811, 986, 120, 271, 218, 171, 649, 57, 1, 389, 928, 971, 475, 843, 781, 274, 882, 690, 405, 264, 969, 784, 470, 485, 369, 345, 376, 680, 839, 859, 275, 30, 404, 450, 481, 989]
Reps: 2112
[30, 30, 40, 45, 48, 95, 114, 119, 120, 120, 133, 135, 141, 145, 151, 170, 170, 171, 190, 197, 204, 209, 218, 227, 237, 242, 250, 260, 264, 271, 273, 274, 274, 275, 292, 293, 330, 334, 345, 361, 368, 369, 376, 382, 389, 396, 404, 405, 409, 418, 450, 470, 475, 476, 481, 485, 489, 495, 498, 514, 551, 566, 571, 610, 634, 643, 649, 660, 678, 680, 684, 690, 702, 713, 719, 775, 779, 781, 784, 800, 805, 811, 812, 834, 839, 843, 846, 859, 882, 889, 904, 908, 910, 926, 928, 929, 969, 971, 986, 989]

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.