

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 «Алгоритмизации»

Выполнила:
Михеева Елена Александровна
2 курс, группа ИВТ-б-з-20-1,
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....

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2023 г.

Тема: Подсчет инверсий с помощью сортировки слиянием.

Порядок выполнения работы.

1. Была написана программа для реализации подсчета количества инверсий в массиве с помощью метода сортировки слиянием. Такой метод работает по принципу «разделяй и властвуй». Он разбивает массив на 2 части и каждую из последующих еще на 2 части и так далее, пока не останутся единичные элементы. Инверсия в данном коде определяется как пара индексов i и j , где $i < j$, но значение элемента с индексом i больше значения элемента с индексом j . Таким образом, алгоритм помимо сортировки также вычисляет количество инверсий в исходном массиве. Программа включает 2 функции: `mergesort()` - рекурсивно разделяет исходный массив пополам, затем сортирует каждую половину отдельно и выполняет их слияние. Возвращает отсортированный массив и общее количество инверсий в массиве; `merge()` - принимает два отсортированных массива и выполняет их слияние, подсчитывая количество инверсий в процессе. Возвращает отсортированный массив после слияния и количество инверсий.

Метод сортировки слиянием с подсчетов инверсий работает за $O(n \cdot \log(n))$.

```
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3  import sys
4
5
6  def mergesort(array):
7      if len(array) <= 1:
8          return array, 0
9      mid = len(array) // 2
10
11      left_list, left_inv = mergesort(array[:mid])
12      right_list, right_inv = mergesort(array[mid:])
13      merged_list, merge_inv = merge(left_list, right_list)
14
15      total = left_inv + right_inv + merge_inv
16
17      return merged_list, total
18
19
20  def merge(left, right):
21      sorted_list = []
22      i, j = 0, 0
23      count_inv = 0
24
25      left_len, right_len = len(left), len(right)
26
27      for _ in range(left_len + right_len):
28          if i < left_len and j < right_len:
29              if left[i] <= right[j]:
30                  sorted_list.append(left[i])
31                  i += 1
32              else:
33                  sorted_list.append(right[j])
34                  count_inv += left_len - i
35                  j += 1
36          elif i == left_len:
37              sorted_list.append(right[j])
38              j += 1
39          elif i == right_len:
40              sorted_list.append(left[i])
41              i += 1
42      return sorted_list, count_inv
43
44
45  if __name__ == "__main__":
46      array = input("Введите массив... ")
47
48      if len(array) <= 1:
49          print("Некорректные данные!", file=sys.stderr)
50          exit(1)
51
52      sorted_array, inversions = mergesort(array)
53
54      print(f"Количество инверсий: {inversions}")
55
```

ПРОБЛЕМЫ ВЫХОДНЫЕ ДАННЫЕ КОНСОЛЬ ОТЛАДКИ ТЕРМИНАЛ ПОРТЫ

Введите массив... 2 31 4 18 7 45 1 0 9
Массив: 2 31 4 18 7 45 1 0 9
Количество инверсий: 11
(Python 3.11.0 Shell)

Рисунок 1. Программа mergesort.py