МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В. И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МО ЭВМ

ОТЧЕТ

по лабораторной работе №2

по дисциплине «Алгоритмы и структуры данных»

Тема: Сортировки

Студент гр. 1381 _____ Возмитель В. Е

Преподаватель _____ Шевская Н. В.

Санкт-Петербург

2022

Цель работы.

Научиться реализовывать сортировку слиянием на конкретной задаче

Задание.

На вход программе подаются квадратные матрицы чисел. Напишите программу, которая сортирует матрицы по возрастанию суммы чисел на главной диагонали с использованием алгоритма сортировки слиянием.

Формат входа:

Первая строка содержит натуральное число n - количество матриц. Далее на вход подаются n матриц, каждая из которых описана в формате: сначала отдельной строкой число *mi* - размерность i-й по счету матрицы. После m строк по m чисел в каждой строке - значения элементов матрицы.

Формат выхода:

Порядковые номера тех матриц, которые участвуют в слиянии на очередной итерации алгоритма. Вывод с новой строки для каждой итерации.

Массив, в котором содержатся порядковые номера матриц, отсортированных по возрастанию суммы элементов на диагонали. Порядковый номер матрицы — это её номер по счету, в котором она была подана на вход программе, нумерация начинается с нуля.

Выполнение работы.

В начале программы написано последовательность команд для считывания параметров ввода: количество матриц, их размер, сами матрицы. Создан класс *Pair* для хранения порядкового номера матрицы и ее суммы элементов на главной диагонали. Основная часть программы — это функция merge, которая реализует сортировку слиянием. Функция принимает список, состоящий из элементов класса Pair, далее рекурсивно делит список пополам, параллельно сравнивая элементы полученных частей. Записывает уже соответсвенные элементы в новый список. В конце функции перезаписываем наш изначальный список, основываясь на новый, и возвращаем его.

Тестирование.

Для проверки работы программы был разработан код тестовой программы.

Всего 3 теста:

- *test_1*. Данный тест был взят с условия лабораторной с сайта. Вход: 3 различные матрицы.

- *test_2*. Тест для проверки работы программы без ввода данных.

- test_3. Тест для проверки особого случая. Вход: 3 матрицы с одинаковой суммой на главной диагонали.

Код файла с тестами находится в приложении А.

Выводы.

Была изучена сортировка слиянием и реализована функции для ее реализации.

ПРИЛОЖЕНИЕ А. ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: algos2.py

```
#python
class Pair:
    def __init__(self, index=0, data=0):
        self.index = index
        self.data = data


def merge(arr):
    answer = []
    if len(arr) <= 1:
        return

    middle = len(arr) // 2
    left, right = arr[:middle], arr[middle:]
    merge(left)
    merge(right)
    index_left = index_right = index_result = 0

    result = [Pair(0, 0) for i in range(len(left) + len(right))]

    while index_left < len(left) and index_right < len(right):

        if left[index_left].data <= right[index_right].data:
            result[index_result].data = left[index_left].data
            result[index_result].index = left[index_left].index
            index_left += 1

        else:
            result[index_result].data = right[index_right].data
            result[index_result].index = right[index_right].index
            index_right += 1

        index_result += 1

    while index_left < len(left):
        result[index_result].data = left[index_left].data
        result[index_result].index = left[index_left].index
        index_left += 1
        index_result += 1

    while index_right < len(right):
        result[index_result].data = right[index_right].data
        result[index_result].index = right[index_right].index
        index_right += 1
        index_result += 1

    for i in range(len(arr)):
        arr[i] = result[i]
```

```
            answer.append(arr[i].index)

        for j in range(len(result)):
            print(result[j].index, end=' ')

        print()

        return answer


    if __name__ == "__main__":
        n = int(input())
        arr = []
        for i in range(n):
            arr.append(Pair(i))
            size = int(input())
            cur_sum = 0

            for j in range(size):
                line = list(map(int, input().split()))
                cur_sum += line[j]

            arr[i].data = cur_sum
        print(*merge(arr))
```

Название файла: test2.py

```
import unittest
from algos2 import *


class TestMethods(unittest.TestCase):
    def test_1(self):
        arr1 = [Pair(0, 32), Pair(1, 11), Pair(2, 3)]
        self.assertEqual(merge(arr1), [2, 1, 0])

    def test_2(self):
        arr2 = []
        self.assertEqual(merge(arr2), None)

    def test_3(self):
        arr3 = [Pair(0, 10), Pair(1, 10), Pair(2, 10)]
        self.assertEqual(merge(arr3), [0, 1, 2])


if __name__ == "__main__":
    unittest.main()
```