**Міністерство освіти і науки України**

**Національний технічний університет України “Київський політехнічний інститут імені Ігоря Сікорського”**

**Факультет інформатики та обчислювальної техніки**

**Кафедра ІПІ**

**Звіт**

з лабораторної роботи № 2 з дисципліни

«Алгоритми та структури даних 2. Структури даних»

„ **Проектування і аналіз алгоритмів внутрішнього сортування**”

**Виконав(ла)** ІП-24 Харечко Олександр Іванович а

**Перевірив Соколовський В.В.**  а

Київ 2023

ЗМІСТ

**1 МЕТА ЛАБОРАТОРНОЇ РОБОТИ ............................................................... 3 2 ЗАВДАННЯ ........................................................................................................ 4 3 ВИКОНАННЯ .................................................................................................... 5 4ВИСНОВОК .....................................................................................................9**

1 МЕТА ЛАБОРАТОРНОЇ РОБОТИ

Мета роботи – вивчити основні принципи роботи методу декомпозиції, пошуку інверсій та закріпити отримані навички на практиці.

2 ЗАВДАННЯ

За допомогою методу декомпозиції розробити алгоритм, який буде розв’язувати наступну задачу.

**Вхідні дані.** Матриця D натуральних чисел розмірності u\*m, де u — це кількість користувачів, m — кількість фільмів. Кожний елемент матриці D[i, j] вказує на позицію фільму j в списку вподобань користувача i. Іншим вхідним елементом є x — номер користувача, з яким будуть порівнюватись всі інші користувачі.

**Вихідні дані.** Список з впорядкованих за зростанням другого елементу пар (i, c), де i — номер користувача, c — число, яке вказує на степінь схожості вподобань користувачів x та c (кількість інверсій).

Розроблена програма повинна зчитувати вхідні дані з файлу заданого формату та записувати дані у файл заданого формату. У вхідному файлі зберігається матриця вподобань всіх користувачів D.

Номер користувача Х, з яким відбувається порівняння всіх інших користувачів, передається аргументом виклику програми через командний рядок.

Вхідний файл представляє собою текстовий файл із U+1 рядків. Перший рядок містить два числа: U та M, де U — кількість користувачів, M — кількість фільмів. Кожен наступний рядок представляє список вподобань (хіт-парад) фільмів відповідних користувачів і містить M+1 число, розділених пробілом. Перше число в рядку є номером користувача (від 1 до U). Решта M чисел є номерами фільмів 1 ,..., M у хіт-параді відповідного користувача.

Вихідний файл представляє також текстовий файл із U рядків. Перший рядок містить одне число — номер користувача Х, з яким відбувалось порівняння всіх інших користувачів. Далі йде U-1 рядків, кожен з яких містить два числа через пробіл: номер користувача i та число c, яке визначає степінь подібності списків вподобань користувачів x та і. Рядки з парами i та c впорядковані за значенням елементу c.

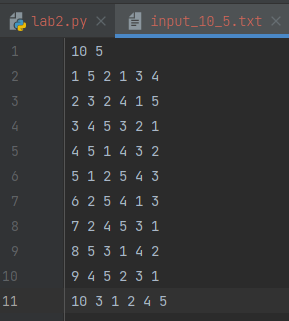
3 ВИКОНАННЯ

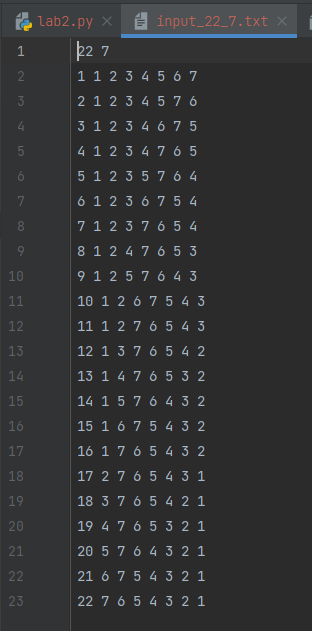
**3.1 Вихідний код**

import sys  
  
  
def merge\_sort(arr):  
 if len(arr) <= 1:  
 return arr, 0  
 mid = len(arr) // 2  
 left, left\_inv = merge\_sort(arr[:mid])  
 right, right\_inv = merge\_sort(arr[mid:])  
  
 merged, merge\_inv = merge(left, right)  
 return merged, merge\_inv + right\_inv + left\_inv  
  
  
def merge(left, right):  
 result = []  
 inversions = 0  
 i = j = 0  
 while i < len(left) and j < len(right):  
 if left[i] <= right[j]:  
 result.append(left[i])  
 i += 1  
 else:  
 result.append(right[j])  
 j += 1  
 inversions += len(left) - i  
  
 result += left[i:]  
 result += right[j:]  
 print(inversions)  
 return result, inversions  
  
  
def inversion(curr, comp):  
 ind = []  
 for i in range(len(comp)):  
 ind.append(comp[curr.index(i + 1)])  
 return merge\_sort(ind)  
  
  
def take\_res(x, data):  
 array = []  
 for i in range(len(data)):  
 if i != x - 1:  
 print("#", i + 1)  
 array.append((i + 1, inversion(data[x - 1], list(data[i][j] for j in range(len(data[i]))))[1]))  
 for i in range(len(array) - 1):  
 for j in range(len(array) - 1 - i):  
 if array[j][1] > array[j + 1][1]:  
 array[j], array[j + 1] = array[j + 1], array[j]  
  
 return array  
  
  
def write(x, res):  
 with open('ip24\_kharechko\_' + str(x) + '\_output.txt', 'w') as f:  
 f.write(str(x))  
 for i in range(len(res)):  
 f.write('\n' + ' '.join(map(str, res[i])))  
  
  
def main():  
 x = int(sys.argv[2])  
 y = str(sys.argv[1])  
 with open(f'data\{y}', 'r') as f:  
 data = [list(map(int, line.strip().split()))[1:] for line in f.readlines()[1:]]  
 result = take\_res(x, data)  
 write(x, result)  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 main()

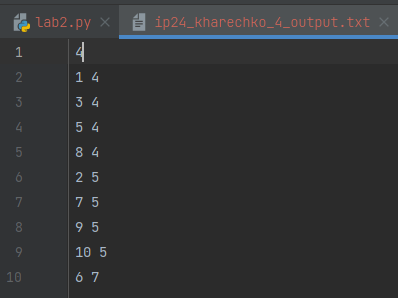
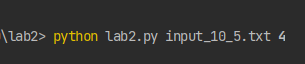
**3.2 Приклад роботи**

Перевіримо роботу програми та розроблених алгоритмів. Створимо як вхідні файли файли input\_10\_5.txt та input\_22\_7.txt, що містить дані 10 та 22 користувачів відповідно. Ці файли мають наступний вигляд:



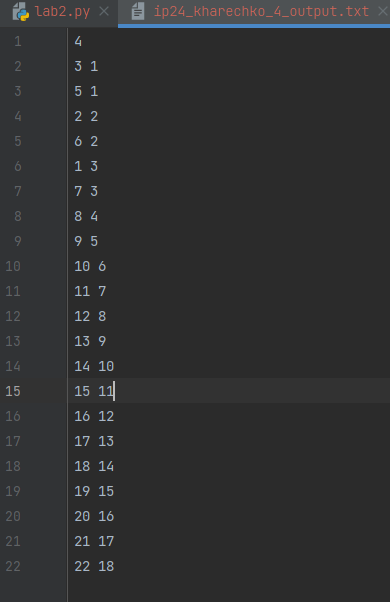


Оберемо файл input\_10\_5.txt та 4-го користувача як користувача для порівняння. Маємо вигляд вихідного файлу:



Оберемо файл input\_22\_7.txt та 4-го користувача як користувача для порівняння. Маємо вигляд вихідного файлу:





ВИСНОВОК

При виконанні лабораторної роботи я дослідив метод декомпозиції, а також пошук інверсій. Ознайомився з прикладом методу декомпозиції: сортуванням злиттям та написав власний модифікований алгоритм цього сортування, що рахує інверсії, розбиваючи масив за принципом “розділяй та володарюй”. Написав власну програмну специфікацію, що працює з файлами та виконує поставлену задачу.