Київський національний університет імені Тараса Шевченка

Факультет комп`ютерних наук та кібернетики

Кафедра інтелектуальних інформаційних систем

Алгоритми та складність

Завдання № 1

“Багатофазне сортування злиттям”

Виконав студент 2-го курсу

Групи К-29

Сукованченко Дмитро Віталійович

2020

**Завдання**

Реалізуйте багатофазне сортування злиттям

**Теорія**

Іноді сортований список виявляється настільки великий, що він не поміщається цілком в пам'ять комп'ютера. В таких випадках доречено використовувати зовнішні сортування, наприклад багатофазне сортування злиттям. Злиттям називається процес об'єднання декількох впорядкованих серій (тобто впорядкованих списків) в одну. Hа кожному кроці ми беpемо найменший з початкових елементів вхідних серій і переміщаємо в кінець вихідний серії.  
 Кожна операція злиття серій, очевидно, вимагає n пересилань елементів, а також до [n/2] порівнянь. (Де n - загальне число елементів серій)

В ідеалі початкові числа серій для багатофазної сортування повинні бути сумами n-1, n-2, ..., 1 послідовних чисел Фібоначчі порядку n -2. Із цього випливає, що наш алгоритм багатофазного злиття застосовний тільки до таких вхідних даних, в яких число серій є сума n-1 таких сум Фібоначі. Якщо число початкових чисел серій не є ідеальною сумою, використовують пусті (фіктивні, холості) серії. Для демонстрації будемо використовувати дві серії.

**Алгоритм**

Маемо всі дані розташовані в одному файлі. Він зчитується і розподіляються по допоміжним файлам. Після чого проводимо злиття в один файл. І так доки при розбитті не виявиться, що файл відсортованний.

При розбитті зчитуем з файла N чисел. В i файл записуемо і найменший елемент з N.

При злитті зчитуемо з кожного допомізного файла число. Найменший з них записуемо у вихідной, а також замість нього зчитуемо нове число з файла.

**Складність**

Кожна операція злиття серій, вимагає n перезаписів елементів (де n - загальне число елементів усіх серій)

Якщо ми маємо N допоміжних файлів. Тоді на кожному кроці відбувається злиття з допоміжних файліа, також ми сортуемо на кожному кроці N елементів, тому число проходів приблизно дорівнює logNn.   
Отож, середня складність О(n logNn)

**Мова програмування**

python3

**Основні модулі програми**

функция сортування  
def f1(input\_file: str, output\_file: str, n) -> None:

...

для розбиття на допоміжні файли   
def split\_files(\*files, input\_file: str) -> bool:

...

для заповнення випадковими числами

def fill\_random(file\_name, left, right, n):

...

для злиття додаткових файлів у вихідних

def merge\_files(\*files, output\_file: str):

...

для взаемодію з користувачем  
def demonstration():

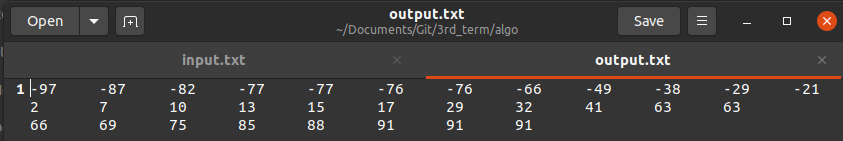
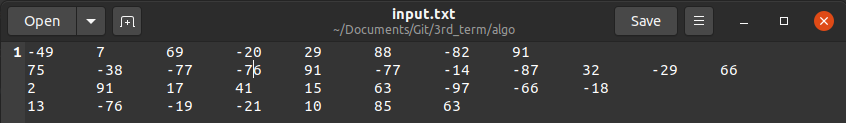
...

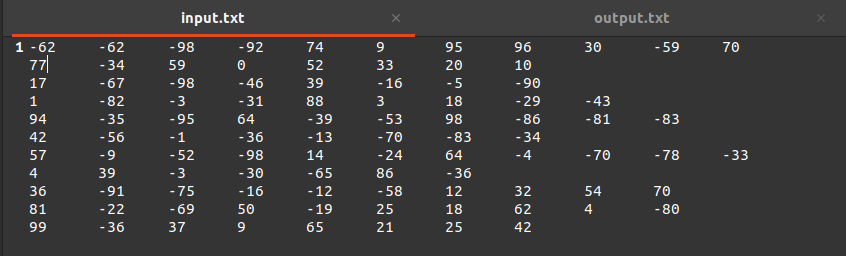
**Інтерфейс користувача**

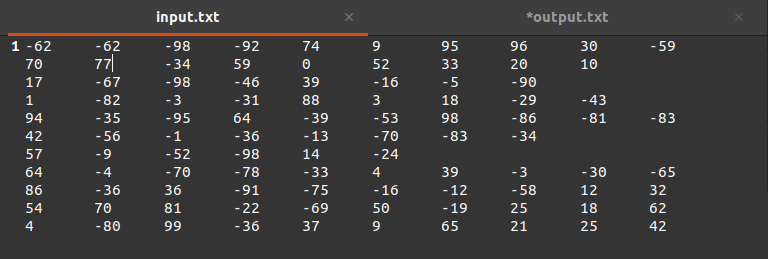
Вхідні дані подаяться з файла input.txt або генеруються випадковим чином, результат роботи виводиться в текстовий файл output.txt.

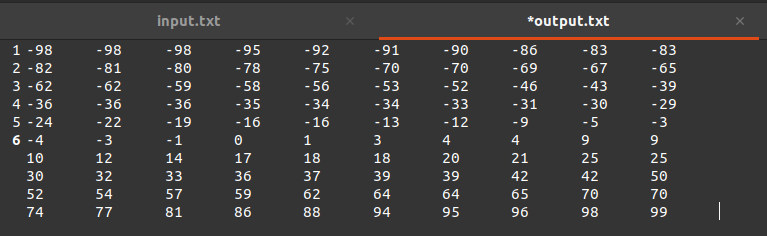
**Тестові приклади**

1.

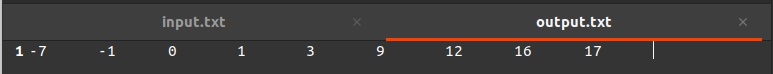
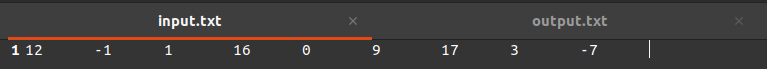


2.





3.



**Висновки**

Коли список, який доводиться сортувати, надто великий, щоб поміститись в оперативну пам’ять комп’ютера, доречно використовувати багатофазне сортування злиттям, складність якого можно оцінити як O(n logn)*.*

**Література**

1. http://algolist.ru/sort/faq/q13.php

2. https://cpct.sibsutis.ru/~artpol/downloads/prog/2012\_2013/s2/prog\_s2\_lect5p4\_v6.pdf (ст. 30-51).

3. https://ozlib.com/868204/informatika/mnogofaznaya\_sortirovka