

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ "ЛЬВІВСЬКА ПОЛІТЕХНІКА"**

**Інститут ІКНІ  
Кафедра ПЗ**

**ЗВІТ**

До лабораторної роботи № 1

**З дисципліни:** *“Алгоритми та структури даних”*

**На тему:** *“Метод сортування бульбашкою”*

**Лектор:**

доц. каф. ПЗ  
Коротєєва Т.О.

**Виконав:**

ст. гр. ПЗ – 22  
Ясногородський Н.В.

**Прийняв:**

асист. каф. ПЗ  
Франко А.В.

« \_\_\_\_ » \_\_\_\_\_ 2022 р.

$\Sigma$  = \_\_\_\_ .

**Тема роботи:** Метод сортування бульбашкою.

**Мета роботи:** Вивчити алгоритм сортування бульбашкою. Здійснити програмну реалізацію алгоритму сортування бульбашкою. Дослідити швидкодію алгоритму.

## ТЕОРЕТИЧНІ ВІДОМОСТІ

Алгоритм сортування бульбашкою (англійською «Bubble Sort») відноситься до класу алгоритмів сортування вибіркою. Алгоритм працює наступним чином — у заданому наборі даних (списку чи масиві) порівнюються два сусідні елементи. Якщо один з елементів не відповідає критерію сортування (є більшим, або ж, навпаки, меншим за свого сусіда), то ці два елементи обмінюються місцями. Прохід по масиву продовжується до тих пір, доки дані не будуть відсортованими. Алгоритм отримав свою назву від того, що процес сортування згідно нього нагадує поведінку бульбашок повітря у резервуарі з водою: найлегша бульбашка піднімається до гори першою. Оскільки для роботи з елементами масиву алгоритм використовує лише порівняння, це сортування на основі порівнянь.

Складність алгоритму у найгіршому випадку дорівнює  $O(n^2)$ , де  $n$  — кількість елементів для сортування. Даний алгоритм має низьку ефективність у випадках, коли  $n$  є досить великим, за винятком рідких конкретних випадків, коли заздалегідь відомо, що масив з самого початку буде добре відсортований.

Одним із засобів підвищення ефективності роботи алгоритму є використання прапорця перестановок, який дозволяє швидше завершити процес сортування.

***Покроковий опис роботи алгоритму сортування бульбашкою з використанням прапорця.***

### **Алгоритм В.**

Задано масив елементів  $R_1, R_2, \dots, R_n$ .

Даний алгоритм реорганізує масив у висхідному порядку, тобто для його елементів буде мати місце співвідношення  $R_i < R_j$  - для всіх  $i, j=1..n$

В1. Цикл за індексом проходження. Повторювати кроки В2 і В3 при  $i=1..n-1$ .

В2. Виконання проходження. Повторювати при  $j=1, 2, \dots, n-i$  : якщо  $R_{j+1} < R_j$  , то встановити  $F=1$  та переставити місцями елементи  $R_j \leftrightarrow R_{j+1}$  ;

В3. Кінець. Вихід.

## ЗАВДАННЯ

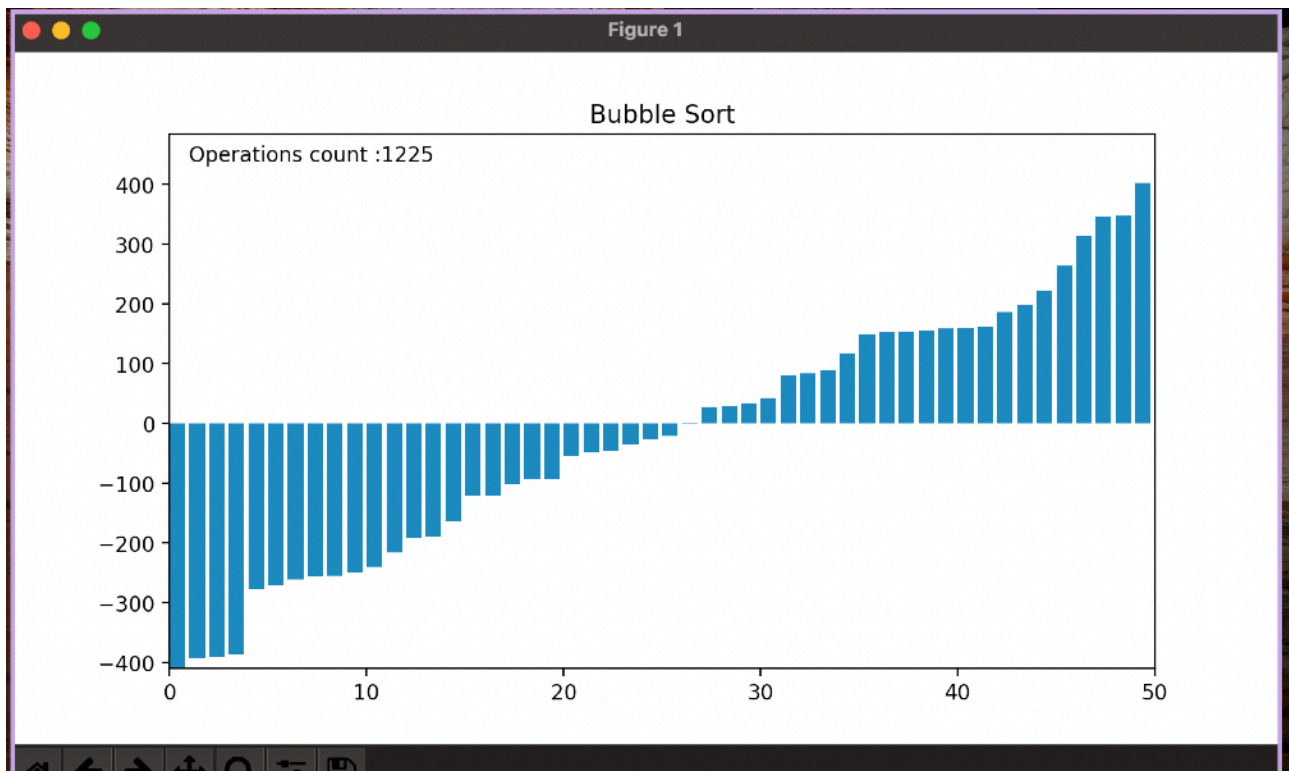
15. *Задано матрицю дійсних чисел. Впорядкувати (переставити) рядки матриці за зростанням суми їх елементів*

## ХІД РОБОТИ

### Функція double sort\_bubble()

```
def swap(A, i, j):  
    A[i], A[j] = A[j], A[i]  
  
def sort_buble(arr, comparator):  
    if len(arr) <= 1:  
        return  
    for i in range(len(arr) - 1):  
        for j in range(len(arr) - 1 - i):  
            if comparator(arr[j], arr[j + 1]):  
                swap(arr, j, j + 1)  
        yield arr
```

## РЕЗУЛЬТАТИ



## ВИСНОВКИ

У процесі виконання лабораторної роботи я проаналізував роботу сортування бульбашкою та реалізував цей алгоритм в ході роботи. Даний алгоритм працює за принципом порівняння сусідніх елементів і заміни їх місцями у випадку виконання заданої умови. Складність алгоритму бульбашки:  $O(n^2)$ .

