**МІНІСТЕРСТВО ОСВІТИ І НАУКИ  УКРАЇНИ**

**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ "ЛЬВІВСЬКА ПОЛІТЕХНІКА"**

Інститут **ІКНІ**

Кафедра **ПЗ**

**ЗВІТ**

До лабораторної роботи № 6

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

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

**Лектор:**

доц. каф. ПЗ

Коротєєва Т.О.

**Виконав:**

ст. гр. ПЗ – 22

Солтисюк Д.А.

**Прийняв:**

асист. каф. ПЗ

Франко А.В.

« \_\_\_\_ » \_\_\_\_\_\_\_\_  2022 р.

 ∑= \_\_\_\_\_                               .

Львів – 2022

**Тема роботи:** Метод сортування підрахунком.

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

**TЕОРЕТИЧНІ ВІДОМОСТІ**

Сортування підрахунком — алгоритм впорядкування, що застосовується при малій кількості різних елементів (ключів) у масиві даних. Час його роботи лінійно залежить як від загальної кількості елементів у масиві так і від кількості різних елементів.

Ідея алгоритму полягає в наступному: спочатку підрахувати скільки разів кожен елемент (ключ) зустрічається в вихідному масиві. Спираючись на ці дані можна одразу вирахувати на якому місці має стояти кожен елемент, а потім за один прохід поставити всі елементи на свої місця.

В алгоритмі присутні тільки прості цикли: довжини N (довжина масиву) та цикл довжини K (величина діапазону). Отже, обчислювальна складність роботи алгоритму становить O(N + K).

В алгоритмі використовується додатковий масив. Тому алгоритм потребує *E*(*K*) додаткової пам’яті. В такій реалізації алгоритм є стабільним. Саме ця його властивість дозволяє використовувати його як частину інших алгоритмів сортування.

***Алгоритм CountingSort***

Дано Rn -масив, n – розмір масиву, i - індекс по масиву, answer – пустий результуючий масив, number\_count –масив для зберігання кількості елементів.

1. Цикл по елементах масиву R, i=1…n. Повторювати CountingSort2.
2. Збільшити значення комірки масиву number\_count з індексом Ri.
3. Цикл по елементах масиву number\_count. Повторювати CountingSort5.
4. Додати значення i до масиву answer               number\_count i разів.
5. Кінець.

**ЗАВДАННЯ**

З двох одновимірних масивів цілих чисел сформувати новий, який включає всі парні числа з першого і непарні з другого масиву. Отриманий масив посортувати в порядку зростання.

**ХІД РОБОТИ**

**Код функції сортування:**

**def** count\_sort(arr, \_):

max\_val = max(arr)

m = max\_val + 1

count = [0] \* m

**for** a **in** arr:

count[a] += 1

**yield** arr

i = 0

**for** a **in** range(m):

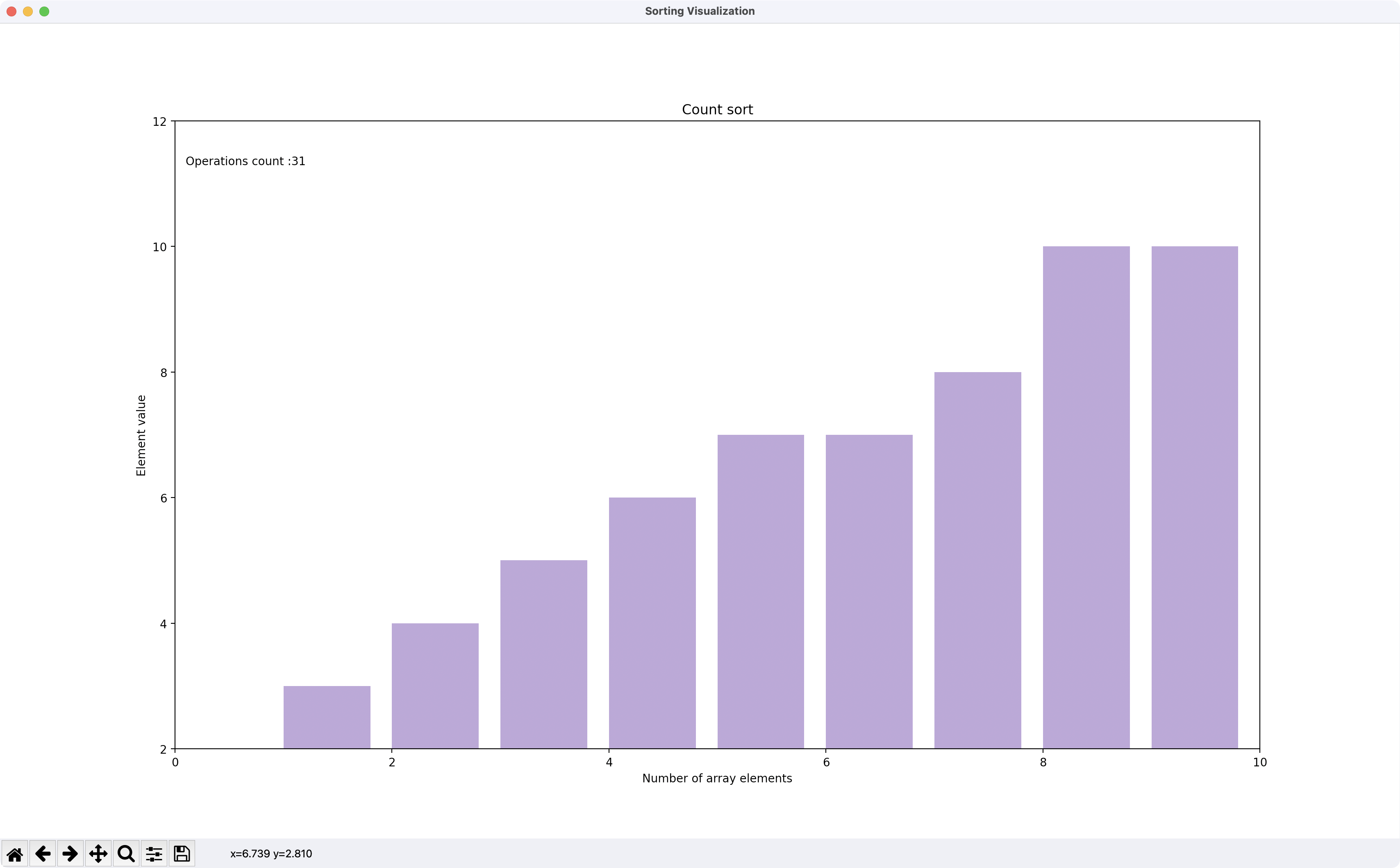
**for** c **in** range(count[a]):

arr[i] = a

i += 1

**yield** arr

**yield** arr

**Візуалізація сортування:**

**ВИСНОВКИ**

Я розглянув реалізацію алгоритму сортування підрахунком. Варто зазначити, що даний алгоритми ефективний лише при малих значеннях вхідних даних і потребує надто багато пам’яті.