**МІНІСТЕРСТВО ОСВІТИ І НАУКИ  УКРАЇНИ**

**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ "ЛЬВІВСЬКА ПОЛІТЕХНІКА"**

Інститут **ІКНІ**

Кафедра **ПЗ**

**ЗВІТ**

До лабораторної роботи № 3

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

**На тему:** *“ Метод сортування Шелла.”*

**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 Шелла.

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

**TЕОРЕТИЧНІ ВІДОМОСТІ**

Сортування Шелла (англійською «Shell Sort») — це алгоритм сортування, що є узагальненням сортування включенням. Його суть полягає в тому, що на кожному кроці групуються та сортуються елементи, що стоять один від одного на певній відстані d. Потім ця відстань зменшується на крок рівний степені двійки. На останньому кроці іде звичайне сортування сусідніх елементів. Перша відстань вибирається відносно кількості елементів в масиві поділена на 2.

Ефективність досягається тим, що кожне наступне впорядкування вимагає меншої кількості перестановок, оскільки деякі елементи вже встали на свої місця. Алгоритм Шелла не є стабільним.

Час роботи алгоритму залежить від вибору значень відстаней *d*.

Існує декілька підходів вибору цих значень:

* При виборі d1=⌊N2⌋, d2=⌊d12⌋, d3=⌊d22⌋,…, dm=1 час роботи алгоритму, в найгіршому випадку, становить *O*(*N*2).
* Якщо *d* — набір чисел виду 3j-12, j ∈N, di<N, то час роботи є *O*(*N*1.5).
* Якщо *d* — набір чисел виду 2i, 3j; i,j ∈N, dk<N, то час роботи є *O*(*N*∙log2*N*).

***Покроковий опис роботи алгоритму сортування вибором.***

**Алгоритм B.**

Задано масив елементів Rn, n – розмір масиву*.* Даний алгоритм реорганізує масив у зростаючому порядку, тобто для його елементів буде мати місце співвідношення: Ri<Ri+1, для всіх i=1…n.

1. h=n/2. Повторюємо кроки 2 - 7. За умови що d>0.
2. i=h. Повторюємо кроки 3 - 6. За умови що   i<n-1.
3. j=i. Повторюємо кроки 4 - 5. За умови що     j>= h та R[j - h]> R[j].
4. Змінюємо місцями R[j] та R[j - h].
5. j=j-h.
6. i=i+1.
7. h=h/2.
8. Кінець. Вихід.

**ЗАВДАННЯ**

Задано двовимірний масив дійсних чисел. Замінити мінімальні елементи кожного рядка на Рисунок 5. Впорядкувати (переставити) стовпці масиву в порядку зростання їх перших елементів.

**ХІД РОБОТИ**

**Код функції сортування:**

**def** shell\_sort(arr, comparator):

sublistcount = len(arr) // 2

**while** sublistcount > 0:

**for** start\_position **in** range(sublistcount):

**yield from** gap\_insertion\_sort(arr, start\_position, sublistcount, comparator)

sublistcount = sublistcount // 2

**def** gap\_insertion\_sort(nlist, start, gap, comparator):

**for** i **in** range(start + gap, len(nlist), gap):

current\_value = nlist[i]

position = i

**while** position >= gap **and** comparator(nlist[position - gap], current\_value):

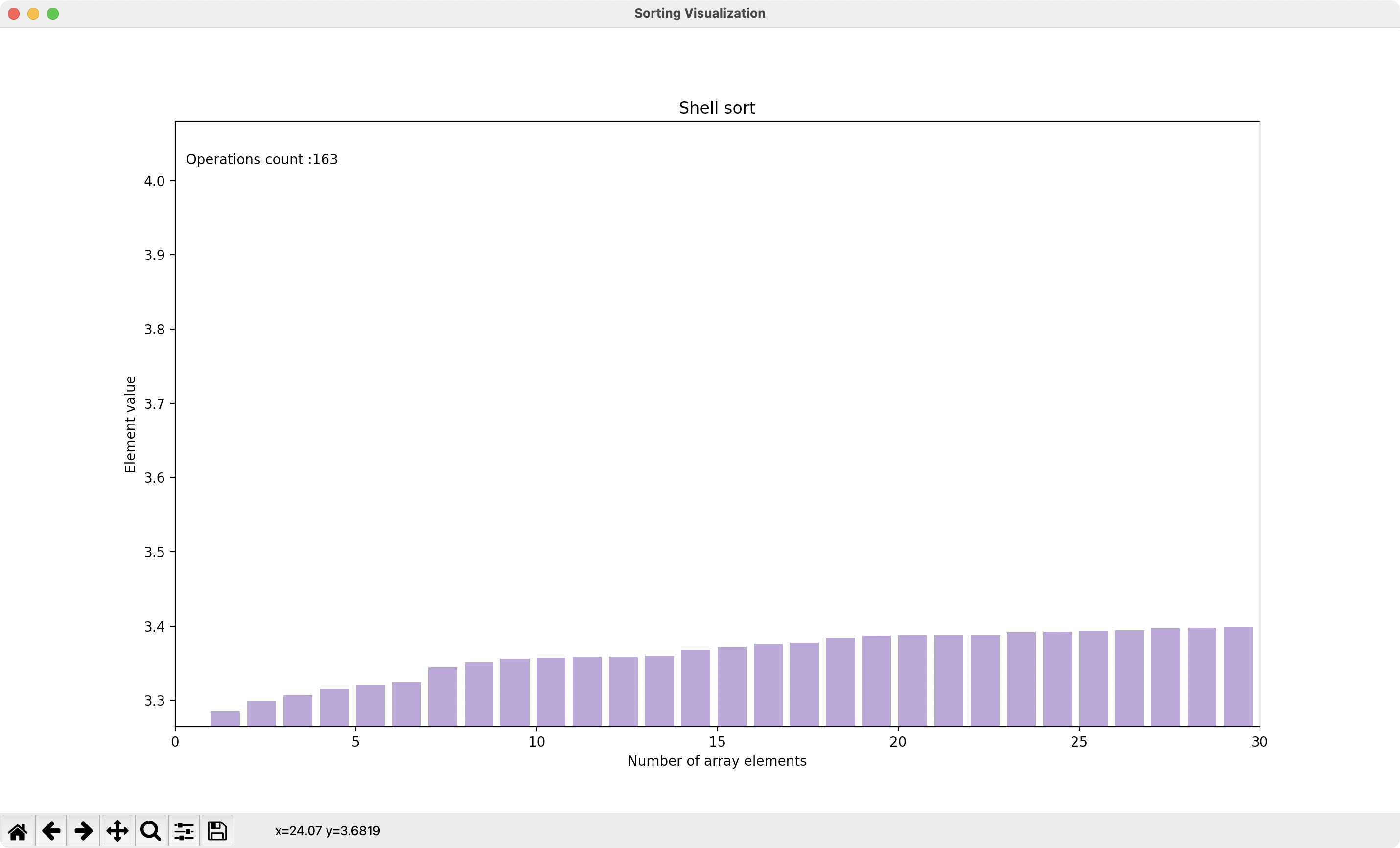
nlist[position] = nlist[position - gap]

position = position - gap

**yield** nlist

nlist[position] = current\_value

**yield** nlist

**РЕЗУЛЬТАТИ**

**ВИСНОВКИ**

На даній лабораторній роботі було розглянуто реалізацію алгоритму сортування Шелла. В результаті виконання лабораторної роботи закріпив навички операцій з масивами та навчився використовувати даний алгоритм для сортування масивів. Варто зазначити, що даний алгоритми не є стабільним, проте буває ефективнішим за сортування бульбашкою. Даний алгоритм не потребує використання додаткової пам’яті. Середня складність алгоритму O(nlog(n)).