

**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
Ясногородський Н.В.

Прийняв:

асист. каф. ПЗ
Франко А.В.

« ____ » _____ 2022 р.

Σ = ____ .

Тема роботи: Метод сортування Шелла.

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

ТЕОРЕТИЧНІ 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 $d_1 = \lfloor N/2 \rfloor$, $d_2 = \lfloor d_1/2 \rfloor$, $d_3 = \lfloor d_2/2 \rfloor, \dots$, $d_m = 1$ час роботи алгоритму, в найгіршому випадку, становить $O(N^2)$.
- Якщо d — набір чисел виду $3j-12$, $j \in \mathbb{N}$, $d_i < N$, то час роботи є $O(N^{1.5})$.
- Якщо d — набір чисел виду 2^i , 3^j ; $i, j \in \mathbb{N}$, $d_k < N$, то час роботи є $O(N \cdot \log^2 N)$.

Покроковий опис роботи алгоритму сортування вибором.

Алгоритм В.

Задано масив елементів R_n , n – розмір масиву. Даний алгоритм реорганізує масив у зростаючому порядку, тобто для його елементів буде мати місце співвідношення: $R_i < R_{i+1}$, для всіх $i = 1 \dots n$.

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 $j \geq h$ 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 одновимірний масив дійсних чисел.

До всіх парних елементів застосувати функцію abs , до непарних - функцію tag .

ХІД РОБОТИ

Код функції сортування:

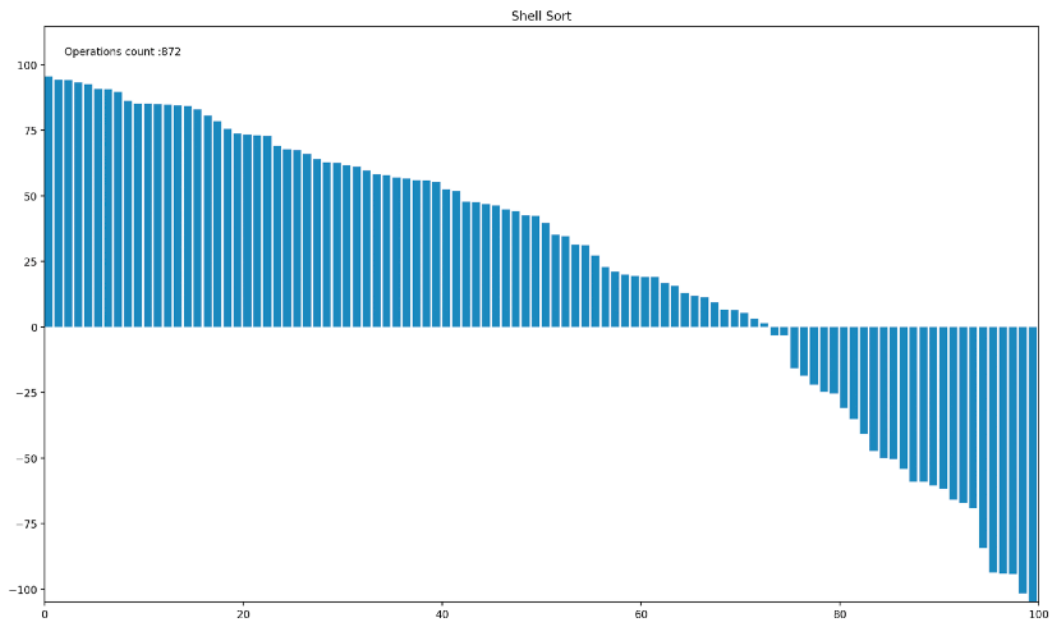
```
def shell_sort(arr, comparator):
    gap = len(arr) // 2
    while gap > 0:
        for start in range(gap):
            yield from gap_insertion_sort(arr, start, gap, comparator)
        gap = gap // 2

def gap_insertion_sort(arr, start, gap, comparator):
    for i in range(start + gap, len(arr), gap):
        current_value = arr[i]

        while i >= gap and comparator(arr[i - gap], current_value):
            arr[i] = arr[i - gap]
            i -= gap
        yield arr

    arr[i] = current_value
    yield arr
```

РЕЗУЛЬТАТИ



ВИСНОВКИ

Під час лабораторної я ознайомився з реалізацією алгоритму сортування Шелла. Варто зазначити, що даний алгоритм не є стабільним. Даний алгоритм не потребує використання додаткової пам'яті. Середня складність алгоритму $O(n \log(n))$.