## Протокол

## Лабораторна робота №2. Варіант 14

### Завдання

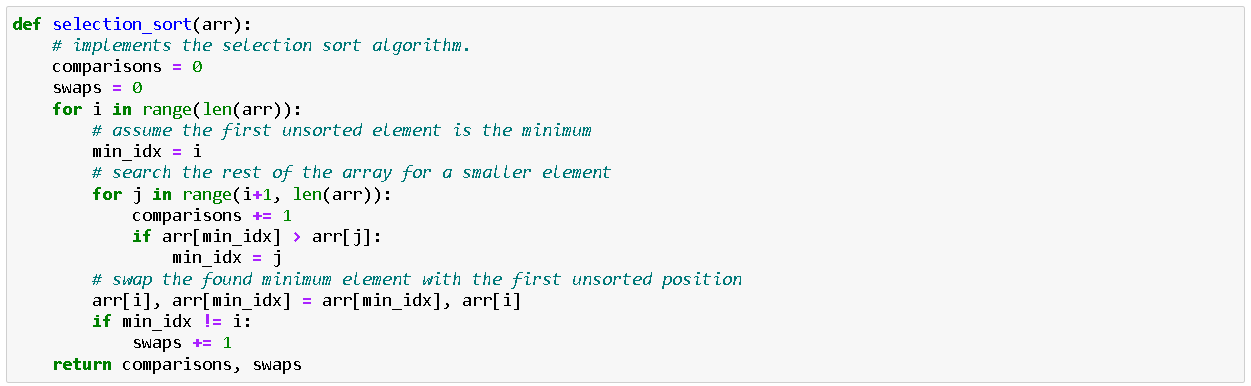
1. Написати програму, що реалізує один з простих методів сортування (згідно з номером варіанту).
2. Написати програму, що реалізує метод Шелла або швидкого соpтування (згідно з номером варіанту).
3. Згенерувати три масиви з випадковими елементами типу **Integer** довжиною 100, 1000 та 10000 елементів, відповідно.
4. Відсортувати одержані масиви за збільшенням елементів, визначивши при цьому такі параметри:
   * кількість порівнянь;
   * кількість обмінів;
   * фактичний час роботи.

### Реалізація

Спочатку генеруємо масив випадкових цілих чисел між -1000 і 1000 заданої довжини.



Функція **selection\_sort(arr)** реалізує алгоритм сортування вибором, де для кожного індексу в масиві шукається мінімальний елемент з невпорядкованої частини та міняється місцями з елементом на цьому індексі.



Функція **sedgewick\_steps(size)** генерує послідовність кроків для методу Шелла за формулою Седжвіка, яка визначає відстані для "проміжного" сортування:



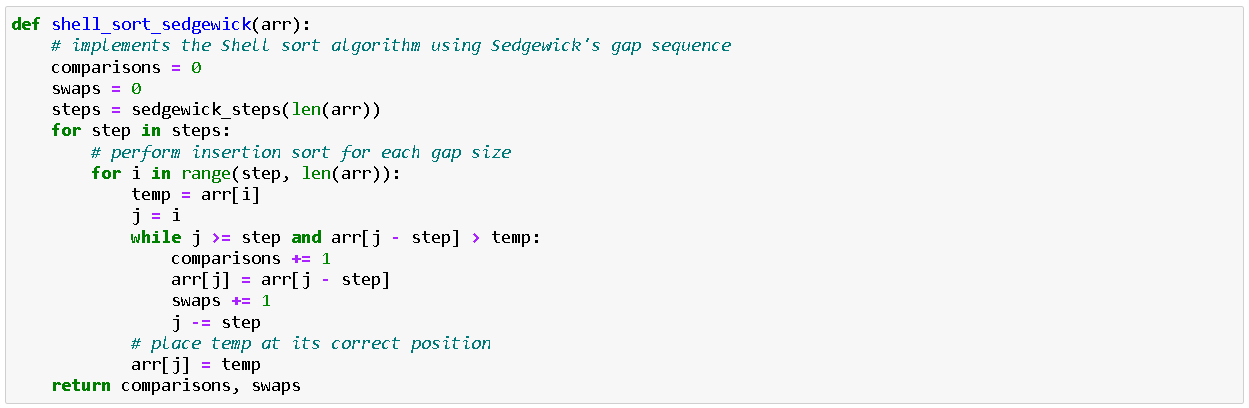
якщо парне;



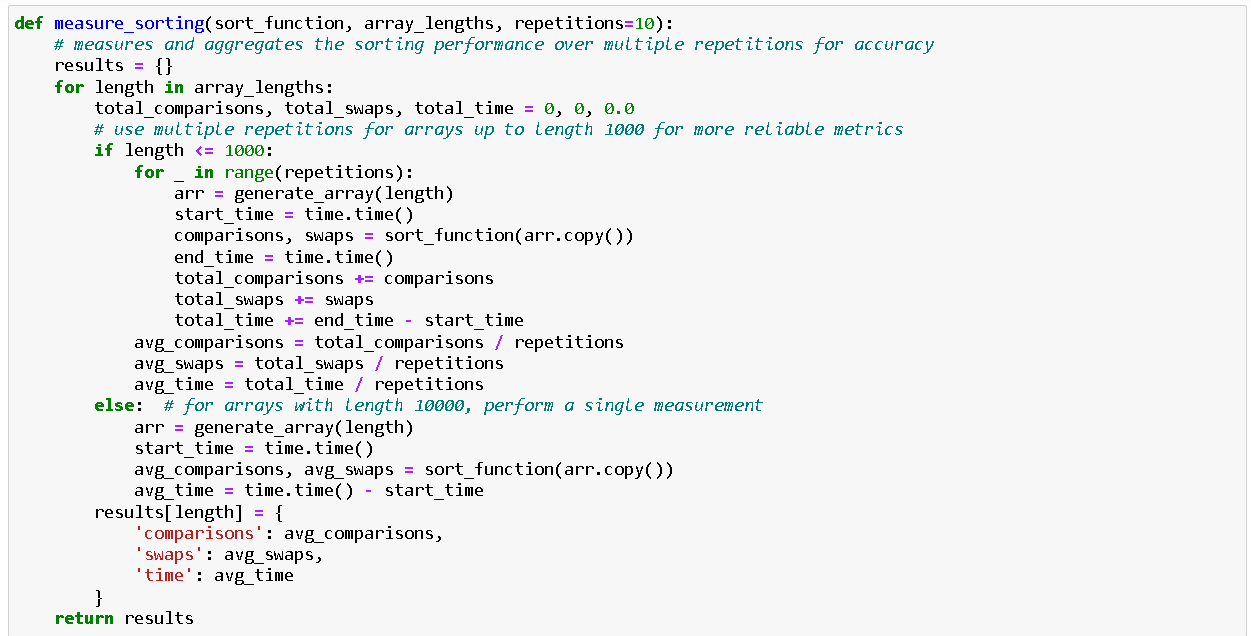
якщо непарне;



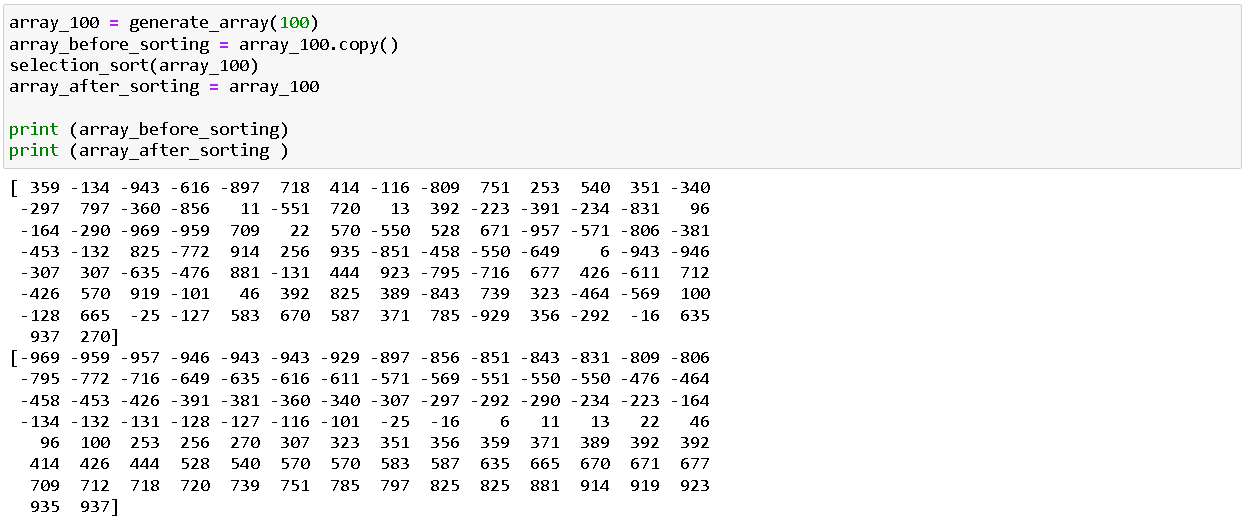
Функція **shell\_sort\_sedgewick(arr)** реалізує алгоритм сортування Шелла з використанням послідовності кроків Седжвіка. Сортування виконується шляхом зменшення відстані між порівнюваними елементами на кожному кроці.



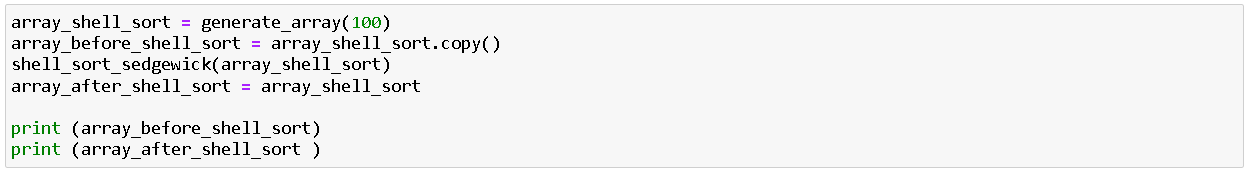
Функція **measure\_sorting(sort\_function, array\_lengths, repetitions=10)** вимірює і збирає статистику ефективності вказаної функції сортування, виконуючи її кілька разів для масивів різних розмірів для отримання більш точних результатів. Вона повертає середні значення кількості порівнянь, обмінів та часу виконання.

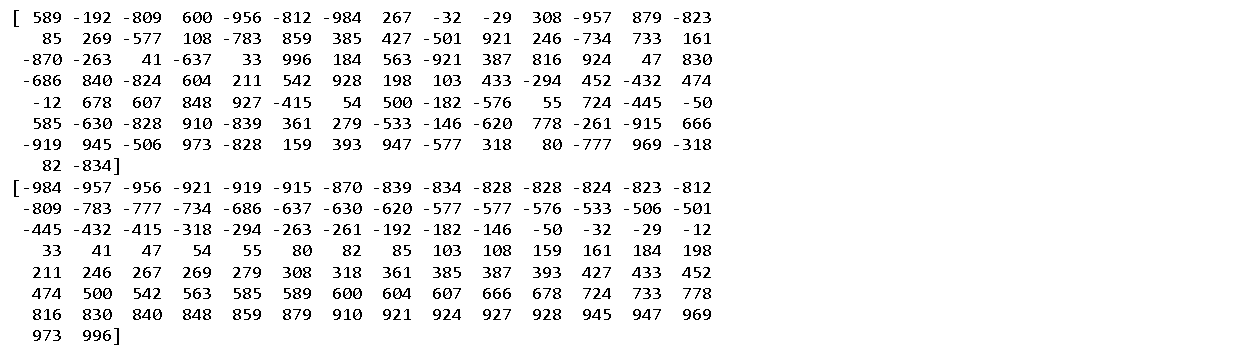


Генерація та сортування масиву методом вибору:

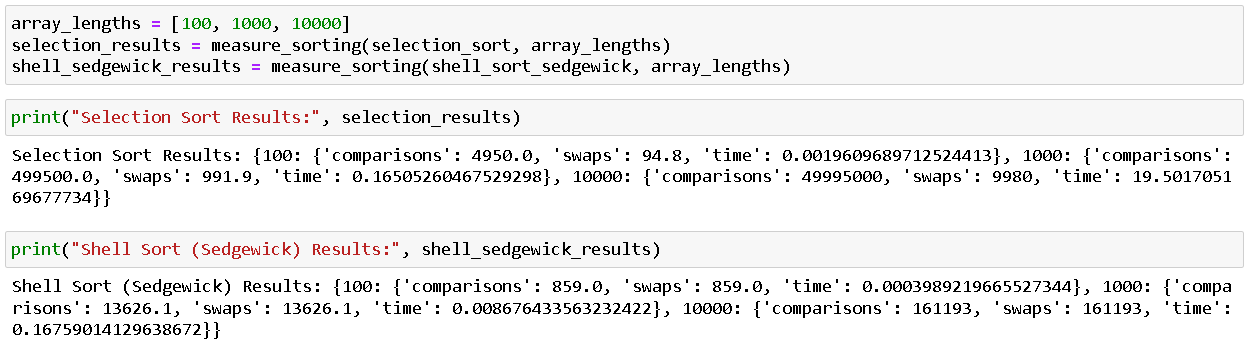


Генерація та сортування масиву методом Шелла:





Визначення параметрів кількості порівнянь та обмінів, а також фактичного часу виконання для методу вибору та Шелла:



|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Результати порiвняння методiв сортування | | | | | | | | | | |
|  | **Метод Вибору** | | | | | **Метод Шелла** | | | | |
| **N** | **К-ть копіювань (M)** | | **К-ть порівнянь (C)** | | **Час (T)** | **К-ть копіювань (M)** | | **К-ть порівнянь (C)** | | **Час (T)** |
| **Теор.** | **Експ.** | **Теор.** | **Експ.** | **Теор.** | **Експ.** | **Теорет.** | **Експ.** |
| 100 | 49 | 94 | 4950 | 4950 | 0.002 | 251 | 782 | 251 | 782 | 0.0003 |
| 1000 | 499 | 992 | 499500 | 499500 | 0.137 | 3981 | 13808 | 3981 | 13808 | 0.006 |
| 10000 | 4999 | 9987 | 49995000 | 49995000 | 15.15 | 63096 | 191491 | 63096 | 191491 | 0.767 |

**Для 100 елементів:**   
  
782/251 = **3.12**

**Для 1000 елементів:**

13808/ 3981 = **3.47**

**Для 10000 елементів:**

191491/63096 = **3.03**

### Аналіз результатів

### Метод Вибору

Кількість порівнянь **(C)**: **n(n−1)/2**

Кількість обмінів **(M)**: (**n−1)/2**

Для методу вибору експериментальні дані дуже близькі до теоретичних оцінок, особливо щодо кількості порівнянь. Експериментальна кількість обмінів дещо менша за теоретичну оцінку, що може бути пояснено наявністю вже впорядкованих сегментів у випадково згенерованих масивах.

### Метод Шелла

Кількість операцій **(C та M)**: **n^1.2**

Метод Шелла виявився значно швидшим за метод вибору, як за кількістю порівнянь та обмінів, так і за фактичним часом виконання. Для великих масивів (10000 елементів) різниця у продуктивності стає ще більш помітною.

Цей аналіз підкреслює важливість вибору правильного алгоритму сортування залежно від розміру даних та вимог до ефективності. Метод Шелла особливо ефективний на великих масивах, а метод вибору може бути прийнятним для невеликих масивів або коли ефективність не є критичним фактором.