

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»**

**Інститут комп'ютерних наук та інформаційних технологій
Кафедра програмного забезпечення**



ЗВІТ

**до лабораторної роботи №3
на тему: «Метод сортування Шелла»
з дисципліни: «Алгоритми і структури даних»**

Лектор:

доц. кафедри ПЗ
Коротєєва Т. О.

Виконав:

ст. гр. ПЗ-22
Чаус О. М.

Прийняв:

асист. кафедри 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 сортування Шелла.

Теоретичні відомості

Сортування Шелла (англійською «Shell Sort») — це алгоритм сортування, що є узагальненням сортування включенням.

Алгоритм базується на двох тезах:

Сортування включенням ефективне для майже впорядкованих масивів.

Сортування включенням неефективне, тому що переміщує елемент тільки на одну позицію за раз.

Тому сортування Шелла виконує декілька впорядкувань включенням, кожен раз порівнюючи і переставляючи елементи, що знаходяться на різній відстані один від одного.

Сортування Шелла не є стабільним.

Сортування Шелла названо на честь автора — Дональда Шелла, який опублікував цей алгоритм у 1959 році.

Покрокове зображення алгоритму

АЛГОРИТМ SH. Сортування масиву алгоритмом Шелла в порядку зростання. **array** – вхідний масив, **i, j** – індекси елементів масиву, **tmp** – тимчасова змінна, **gap** – проміжок між елементами

SH1. Ініціалізація **gap** = $N/2$.

SH2. Ініціалізація **i** = **gap**

SH3. Ініціалізація **j** = **i**, **tmp** = **array[i]**.

SH4. **array[j] = array[j - gap]**. Перехід до **SH5**.

SH5. **j = j - gap**. Якщо **j ≥ gap** і **array[j - gap] > tmp**, перехід до **SH4**, інакше перехід до **SH6**.

SH6. **array[j] = tmp**.

SH7. **i = i + 1**. Якщо **i < gap**, перехід до **SH3**, інакше перехід до **SH7**.

SH7. **gap = gap/2**. Якщо **gap > 0**, перехід до **SH2**, інакше перехід до **SH8**.

SH8. Вихід з алгоритму.

Індивідуальне завдання

Написати віконний додаток на мові програмування C або C++. Реалізована програма повинна виконувати наступну послідовність дій:

- 1) запитуватиме в користувача кількість цілих чотирьохбайтових знакових чисел — елементів масиву, сортування якого буде пізніше здійснено;
- 2) виділятиме для масиву стільки пам'яті, скільки необхідно для зберігання вказаної кількості елементів, але не більше;
- 3) ініціалізовуватиме значення елементів масиву за допомогою стандартної послідовності псевдовипадкових чисел;
- 4) засікатиме час початку сортування масиву з максимально можливою точністю;
- 5) сортуватиме елементи масиву в неспадному порядку за допомогою алгоритму сортування вибором;
- 6) засікатиме час закінчення сортування масиву з максимально можливою точністю;
- 7) здійснюватиме перевірку упорядкованості масиву;
- 8) повідомлятиме користувачу результат перевірки упорядкованості масиву та загальний час виконання сортування з максимально можливою точністю;

9) звільнятиме усю виділену раніше пам'ять.

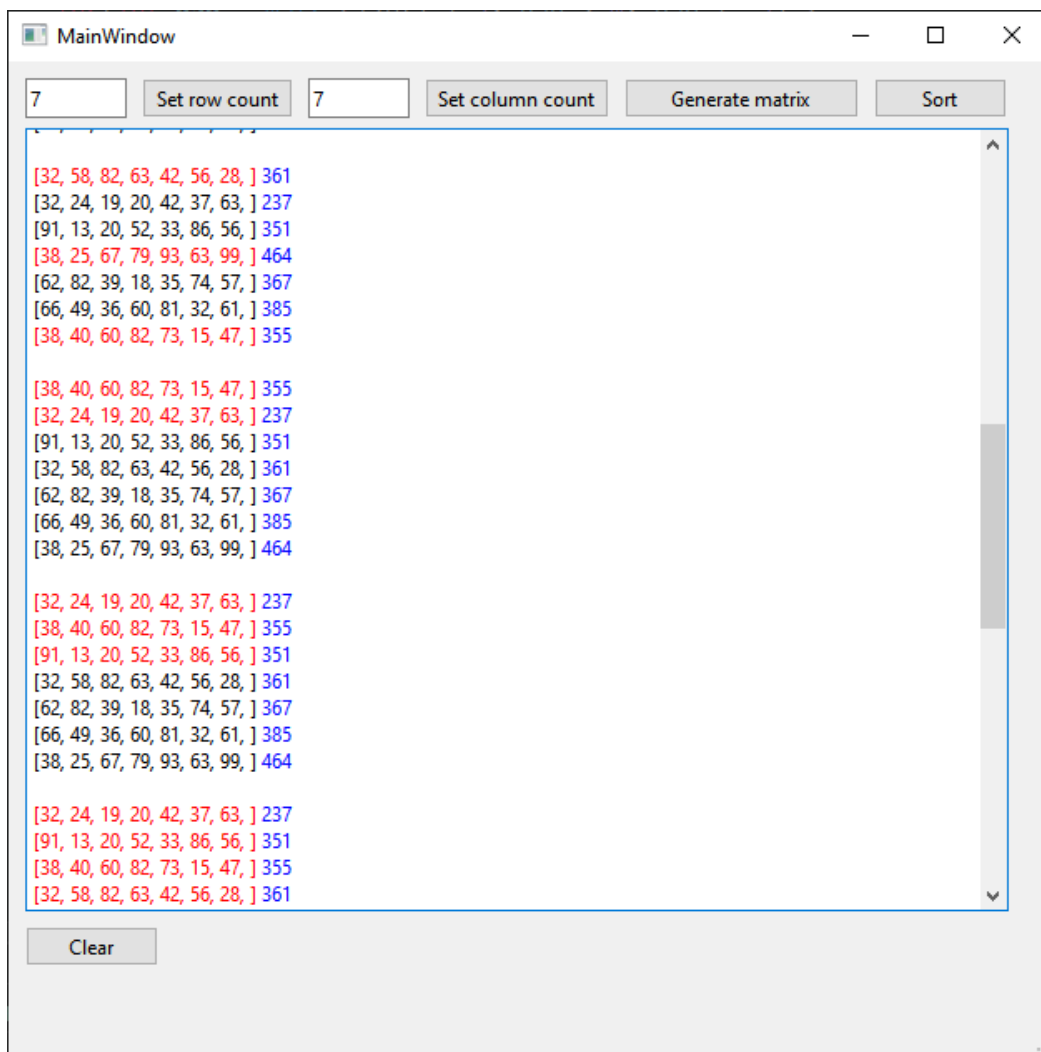
Варіант 15:

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

Код програми

```
...  
void MainWindow::shell_sort(std::vector<std::vector<int>> &matr) {  
    for(int gap = matr.size() / 2; gap > 0; gap /= 2) {  
        for(std::size_t i = gap; i < matr.size(); i++) {  
            auto temp_row = matr[i];  
            int temp_val = std::accumulate(matr[i].begin(), matr[i].end(), 0);  
            std::size_t j = i;  
            while( (j >= gap) && (std::accumulate(matr[j - gap].begin(), matr[j  
- gap].end(), 0) > temp_val) ) {  
                matr[j] = matr[j - gap];  
                j -= gap;  
            }  
            matr[j] = temp_row;  
        }  
    }  
}  
}
```

Скріншоти програми



Висновок: під час виконання лабораторної роботи вивчив алгоритм сортування Шелла та здійснив програмну реалізацію. Дослідив швидкодію алгоритму Шелла, що дорівнює $O(n^{1.5})$