**Практичне завдання №1.**

Порівняння роботи Selection sort,

Insertion sort, Shellsort

Виконав:

Михайло Броницький

Студент групи ПКН-20-Б

**Завдання**

Порівняти ефективність роботи чотирьох алгоритмів: Selection sort, Insertion sort, Merge sort, Shellsort для наступних вхідних даних:

1. випадковим чином згенерований масив (згенерувати 5 експериментів та записати середнє значення) ;
2. значення масиву відсортовані у порядку зростання;
3. значення масиву відсортовані у порядку зменшення;
4. масив містить лише елементи з множини {1, 2, 3} - тобто в масиві багато елементів, які повторюються; згенерувати 3 експерименти (шляхом перестановки значень масивів) та записати середнє значення.

Провести експерименти на розмірах масивів, які починаються від 27 і до 215, з кроком *х2* (тобто кожний раз розмір масиву збільшуємо вдвічі). Всі чотири алгоритми проганяються на одних і тих самих масивах, тобто:  
  
- Для номеру експерименту від 1 до 4:  
-- Для розміру масиву від 27 і до 215:  
--- Згенерувати масив *A*  
--- Для типу алгоритму в { Selection sort, Insertion sort, Merge sort, Shellsort }:  
---- Запустити алгоритм на згенерованому масиві *A*  
---- Запам’ятати час роботи алгоритму на згенерованому масиві  
---- Запам’ятати кількість операцій порівняння під час запуску алгоритму

1. Випадковим чином згенерований масив
2. Значення масиву відсортовані у порядку зростання
3. Значення масиву відсортовані у порядку спадання
4. Масив містить лише елементи з множини {1, 2, 3}

Аналіз алгоритмів

Selection sort, Insertion sort, Merge sort, Shellsort

У цій роботі було проведено експерименти з 4 різними алгоритмами сортування на 4 різних масивах. Як не дивно найкраще себе проявив Shellsort у якого складність О(n^2). Але на малих та середніх розмірах даних вір поводить себе як О(n\*log(n)). А через малу константу при алгоритмічній складності він переганяє навіть Merge sort, який є 2-им за швидкістю роботи. Іншою перевагою над Shellsort є те, що він не потребує додаткової пам’яті на відміну від Merge sort.

Але на значно більших даних Merge sort повинен перегнати Shellsort по продуктивності, оскільки прямі роботи обох алгоритмів наближаються(на випадкових даних це і сталось).

Щодо Insertion sort та Selection sort то в них складність O(n^2) і вони є найповільнішими. На відсортованих даних у порядку спадання Insertion sort програє, бо це найгірший випадок для нього. Однак на випадкових даних він перевершує Selection sort, через те, що Insertion sort не пересортовує уже відсортовані частинки даних (які утворюються випадковим чином), а Selection sort по суті виконує повний перебір усіх n^2 варіантів, на будь-яких типах даних. А на відсортованих даних у порядку зростання Insertion sort взагалі перевершує решту алгоритмів так як тоді його складність є O(n). Це означає що його найвигідніше використовувати для сортування майже відсортованих масивів.