

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

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



**ЗВІТ**

**до лабораторної роботи №5**  
**на тему: «Метод сортування злиттям»**  
**з дисципліни: «Алгоритми і структури даних»**

**Лектор:**

доц. кафедри ПЗ  
Коротеева Т. О.

**Виконав:**

ст. гр. ПЗ-22  
Чаус О. М.

**Прийняв:**

асист. кафедри ПЗ  
Франко А. В.

« \_\_\_\_ » \_\_\_\_\_ 2022 р.

$\Sigma$  = \_\_\_\_ .....

**Тема роботи:** Метод сортування злиттям.

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

### Теоретичні відомості

Сортування злиттям (англійською «Merge Sort») — алгоритм сортування, в основі якого лежить принцип «розділяй та володарюй» (англійською «Divide and Conquer»). В основі цього способу сортування лежить злиття двох упорядкованих ділянок масиву в одну впорядковану ділянку іншого масиву.

Під час сортування в дві допоміжні черги з основної поміщаються перші дві відсортовані підпоследовності, які потім зливаються в одну і результат записується в тимчасову чергу. Потім з основної черги беруться наступні дві відсортовані підпоследовності і так до тих пір доки основна черга не стане порожньою. Після цього последовність з тимчасової черги переміщується в основну чергу. І знову продовжується сортування злиттям двох відсортованих підпоследовностей. Сортування триватиме до тих пір поки довжина відсортованої підпоследовності не стане рівною довжині самої последовності.

Сортування злиттям можна задати рекурсивно: масив поділяється на дві приблизно рівні частини, які після сортування (тим самим способом) зливаються. Коли ж довжина частини масиву зменшується до 1, відбувається повернення з рекурсії. Завершуючи описання сортування злиттям, скажемо, що цей алгоритм є першим із ефективних алгоритмів сортування. У 1945 році його винайшов Джон фон Нейман, один із піонерів програмування.

### Покрокове зображення алгоритму

**АЛГОРИТМ MS.** Сортування масиву алгоритмом злиття в порядку зростання. **array** – вхідний масив, **i, l, r** – індекси елементів масиву та його підмасивів, **middle** – середина масиву, **left, right** підмасиви.

**QS1.** Виклик ф-ції з масивом **array**.

**QS2.** Якщо довжина **array** > 1, перехід на **QS3**. Інакше перехід на **QS9**.

**QS3.** Ініціалізація **middle** = середина **array**. **left** = **array[:middle]**, **right** = **array[middle:]**

**QS4.** Перехід на **QS1**, де **array** = **left**, Перехід на **QS1**, де **array** = **right**

**QS5.** Ініціалізація **i** = **l** = **r** = 0.

**QS6.** Якщо **l** < розмір **left** і **r** < розмір **right** перехід на **QS7**, інакше перехід на **QS8**

**QS7.** Якщо **left[l]** < **right[r]**, **array[i]** = **left[l]**, **l++**, інакше **array[i]** = **right[r]**, **r++**. **i++**, перехід на **QS6**.

**QS8.** Якщо **l** < розмір **left**, перехід на **QS9**, інакше перехід на **QS10**.

**QS9.** **array[i]** = **left[l]**, **l++**, **i++**. Перехід на **QS8**.

**QS10.** Якщо **r** < розмір **right**, перехід на **QS9**, інакше перехід на **QS12**.

**QS11.** **array[i]** = **right[r]**, **r++**, **i++**. Перехід на **QS10**.

**QS12.** Вихід з функції

### Індивідуальне завдання

Написати віконний додаток на мові програмування C або C++. Реалізована програма повинна виконувати наступну последовність 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 17:

Задано одновимірний масив дійсних чисел. До всіх парних елементів застосувати функцію  $\tan x - x$ , до непарних - функцію  $|x|$ . Отриманий масив посортувати в порядку спадання.

#### Код програми

```
import numpy as np
def merge_sort(array):
    for idx, val in enumerate(array):
        if val % 2:
            array[idx] = np.abs(val)
        else:
            array[idx] = (np.tan(val) - val)
    def ms(array):
        def merge(left_array, right_array, array):
            i, l, r = 0, 0, 0
            while l < left_array.size and r < right_array.size:
                if left_array[l] < right_array[r]:
                    array[i] = left_array[l]
                    l += 1
                    i += 1
                else:
                    array[i] = right_array[r]
                    r += 1
                    i += 1
            while l < left_array.size:
                array[i] = left_array[l]
                i += 1
                l += 1
            while r < right_array.size:
                array[i] = right_array[r]
                i += 1
                r += 1
        if array.size > 1:
            middle = array.size // 2
            left_array = np.copy(array[:middle])
            right_array = np.copy(array[middle:])
            ms(left_array)
            ms(right_array)
            merge(left_array, right_array, array)
    ms(array)
```

## Скріншоти програми

Figure 1

— □ ×

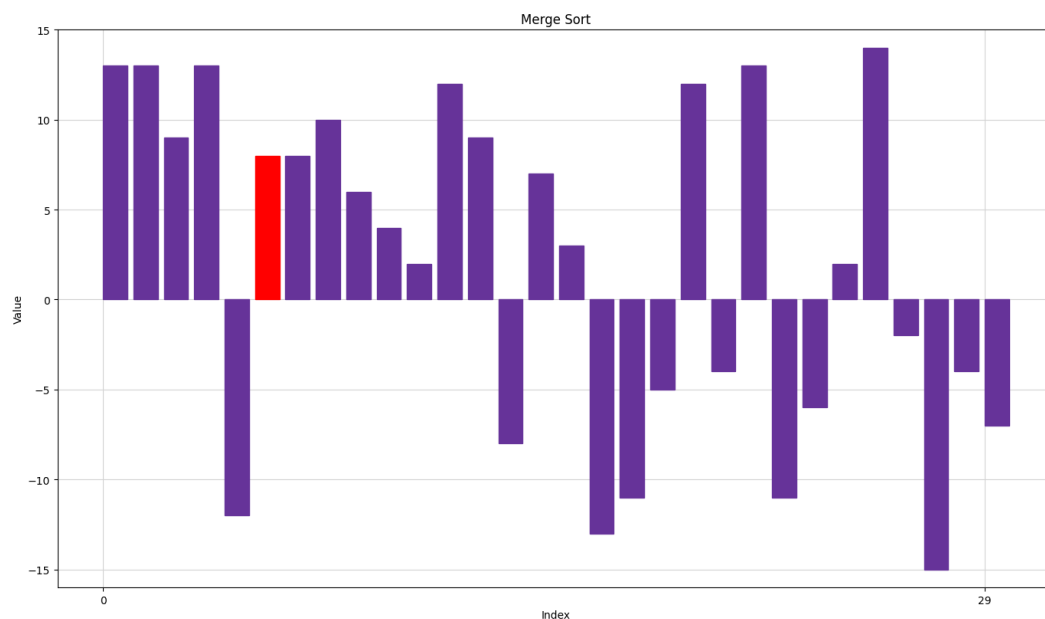


Figure 1

— □ ×

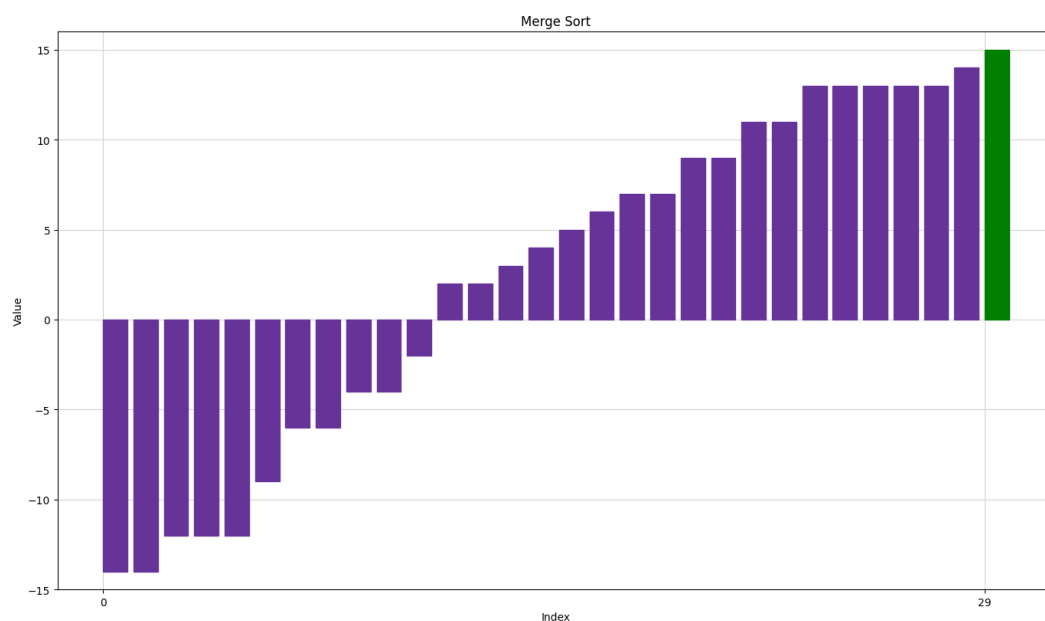


Figure 1

x=16.28 y=-10.52

**Висновок:** під час виконання лабораторної роботи вивчив алгоритм злиття та здійснив програмну реалізацію. Швидкодія алгоритму дорівнює  $O(n \log n)$