

# Merge Sort

Dadmo

November 25, 2024

## 1 Plan

1. Merge Sort
2. Random Lists
3. Linked Lists
4. Converters
5. Comparisions and Results

## 2 Merge Sort

MergeSort — клас, який має реалізовані 4 варіанти алгоритму сортування злиттям.

1. Рекурсивний (Top-Down MergeSort)
2. Ітеративний (Bottom-Up MergeSort)
3. Ітеративний з оптимізаціями cutoff(-to-insertion), stop-if-already-sorted, eliminate-the-copy-to-the-auxiliary-array.
4. Сортування злиттям для зв'язного списку

В моїй реалізації Ітеративного сортування злиттям використовується менше порівнянь для реверсивно відсортованих списків.

В моїй реалізації Ітеративного сортування злиттям з оптимізаціями масиви розміру між  $80eA$  та  $90eA$ , де  $eA$  —  $10$  в степені  $A$ , сортуються не до кінця (Останні  $e$  елементів) через оптимізацію eliminate-the-copy-to-the-auxiliary-array.

Також реалізовано порівняльний аналіз (з даними різного розміру) всіх чотирьох варіантів алгоритму сортування злиттям відносно часу виконання, кількості проведених порівнянь, операцій "копіювань" та використаної пам'яті. Окрім цього є показ на скільки відсотків обрахунки вже завершено.

### 3 Random Lists

RandomLists — клас, який має реалізовані 5 варіанти генерації списків.

1. Повністю відсортований (sorted) — на вхід подається лише розмір списку.
2. Випадкові (random) — на вхід подається лише розмір списку.
3. Майже відсортований (almostsorted) — на вхід подається розмір списку, та відсоток безпорядку.
4. Відсортовані в зворотному порядку (reverse) — на вхід подається лише розмір списку.
5. Лише з декількома різними значеннями (somenumbers) — на вхід подається розмір списку, та діапазон значень (Початок, Кінець).

### 4 Linked Lists

Node — клас, який зберігає дані, а також посилання на наступний та попередній елементи. Для нього реалізовано отримання розміру в байтах, а також реалізовані усі порівняння.

LinkedList — клас, який зберігає посилання на перший та останній елементи. Для нього реалізовано отримання розміру в байтах, а також реалізоване додавання елементів, розширення списку, підрахунок певної підкількості елементів, а також видалення елементів.

Додатково реалізовано призначення та отримання елемента за індексом, слайсом індексів списку.

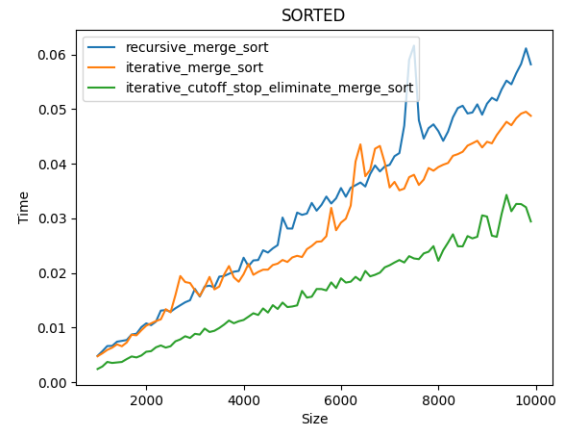
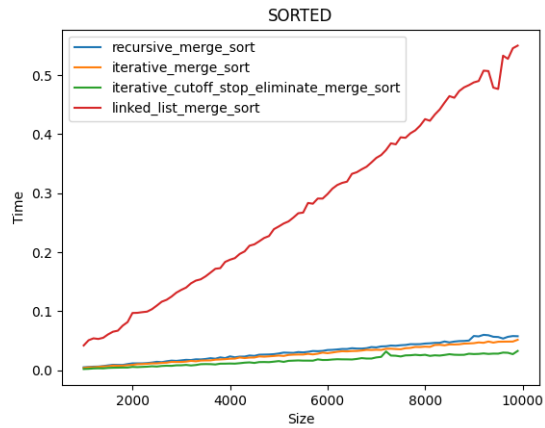
### 5 Converters

Converter — клас, в якому реалізовано конвертацію масиву у зв'язний список та навпаки.

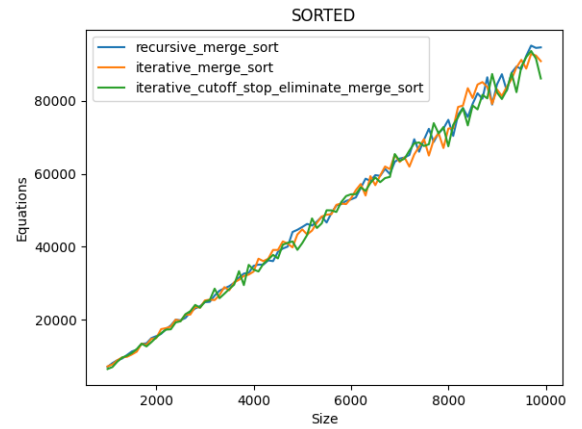
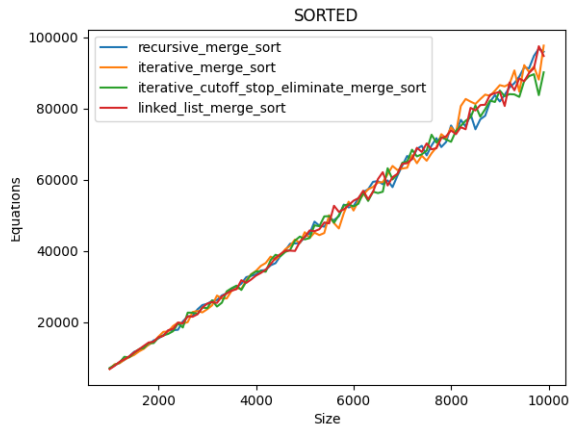
### 6 Comparisions and Results

Подивимось результати виконання. Розміри масивів: від 10 до 10000 елементів з кроком в 100.

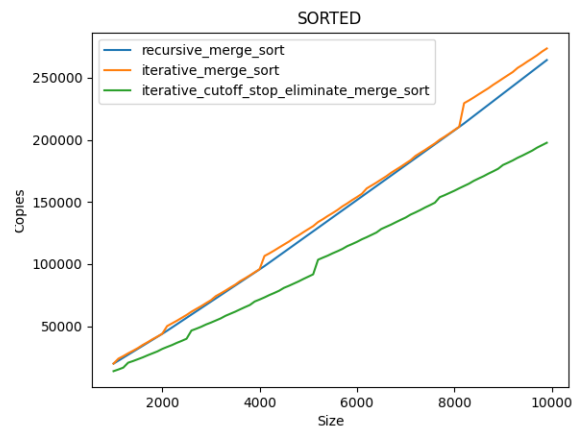
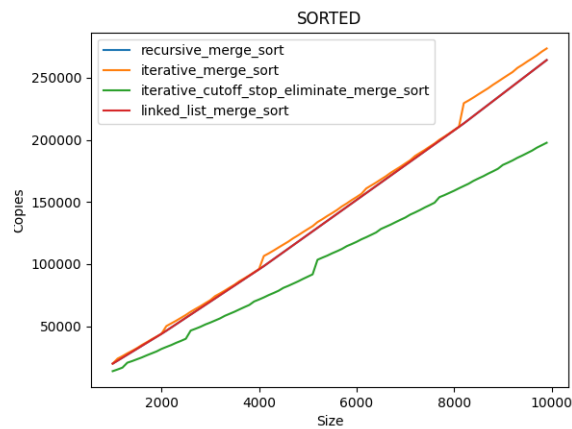
Спочатку для відсортованих масивів:  
Час виконання:



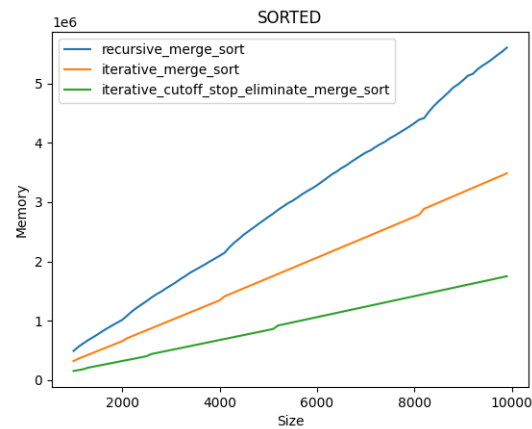
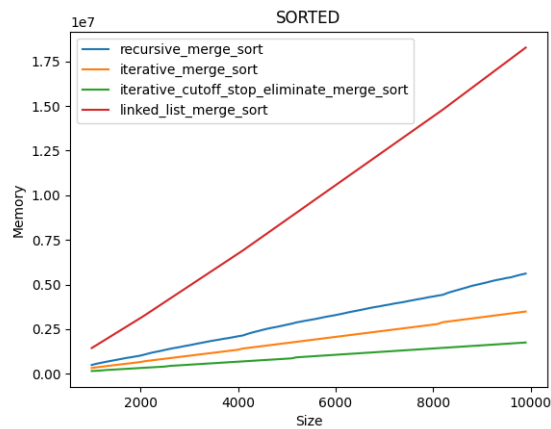
Кількість проведених порівнянь:



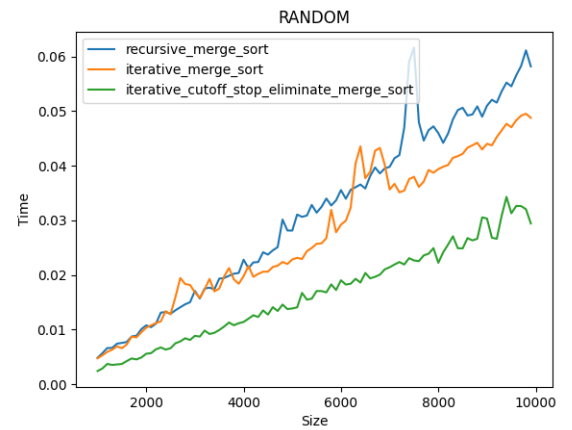
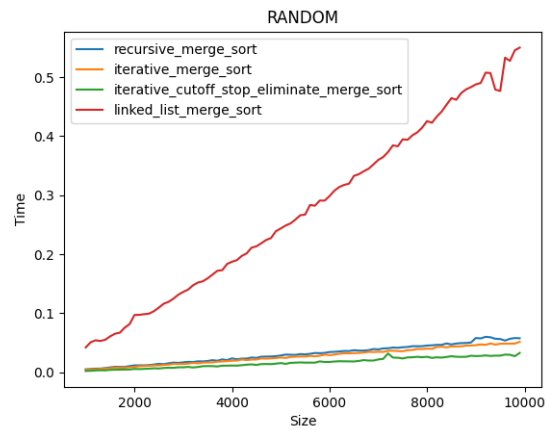
Операції "копіювань":



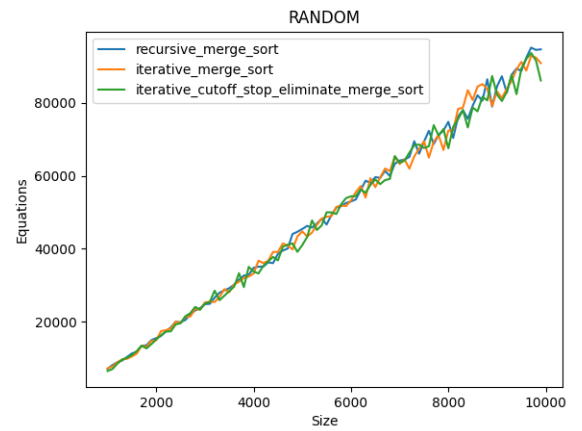
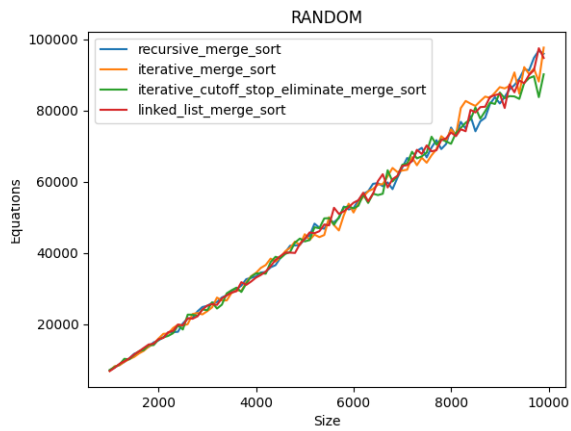
Використано пам'яті:



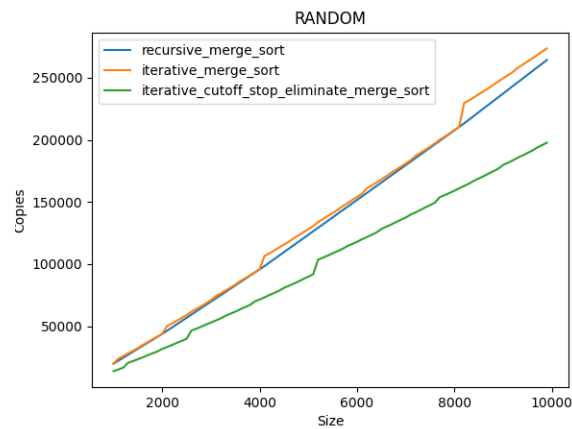
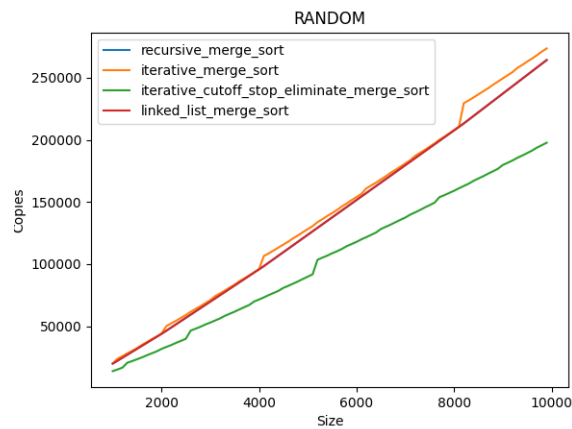
**Потім для масивів випадкових значень:**  
**Час виконання:**



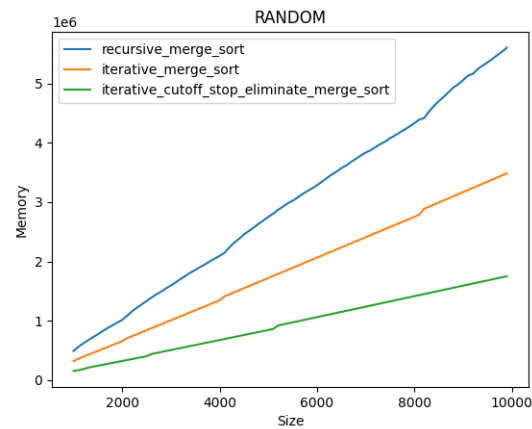
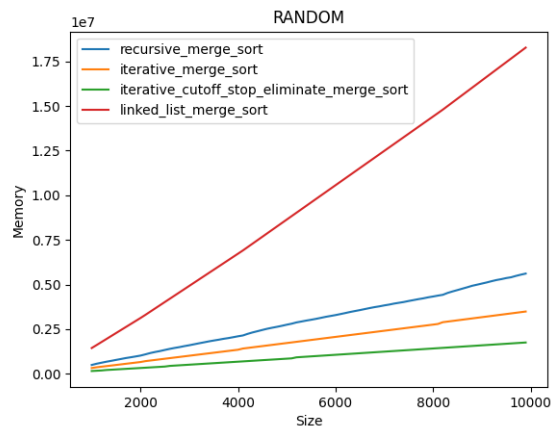
**Кількість проведених порівнянь:**



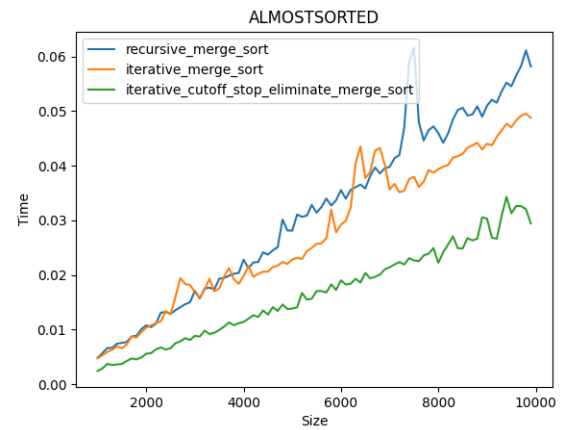
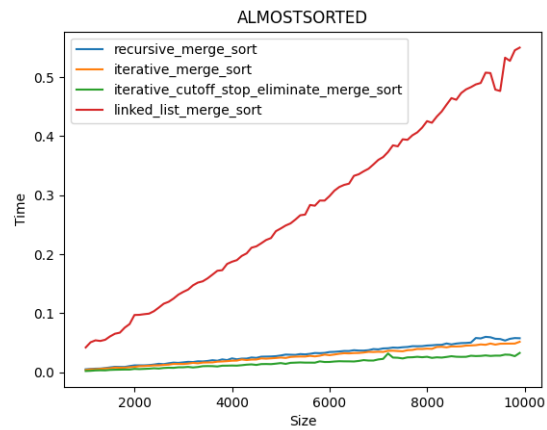
Операції "копіювань":



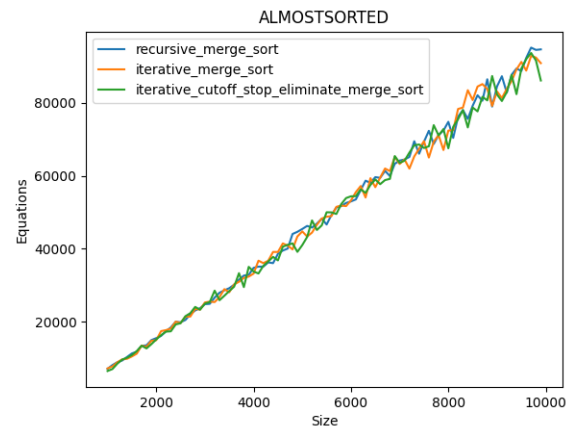
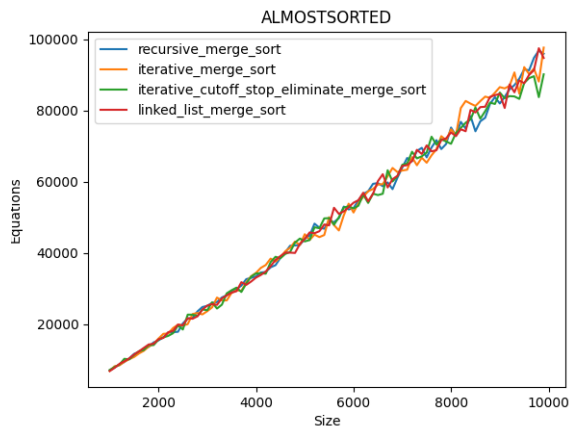
Використано пам'яті:



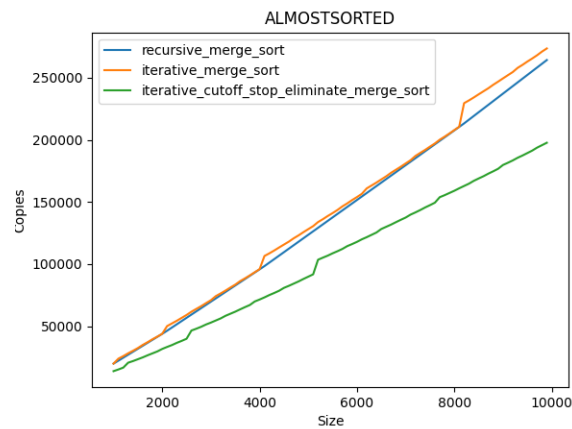
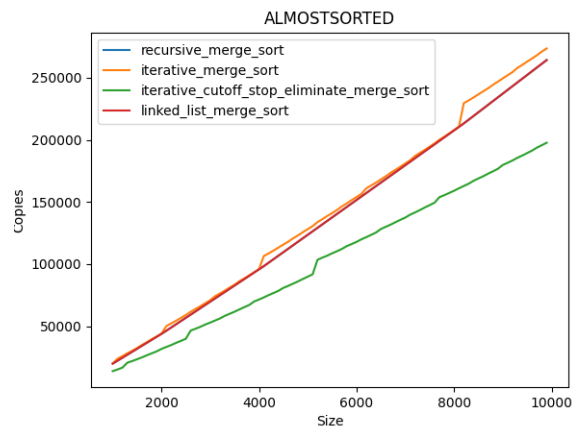
**Масив майже відсортованих значень:**  
Час виконання:



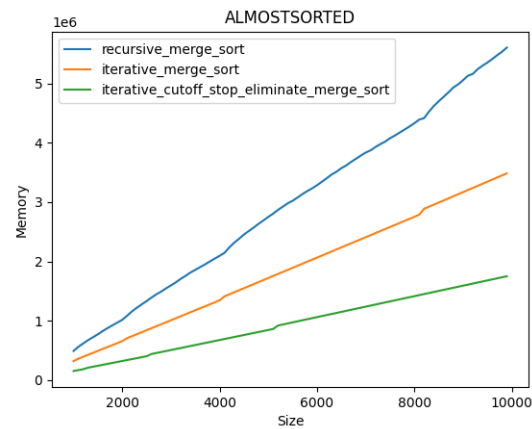
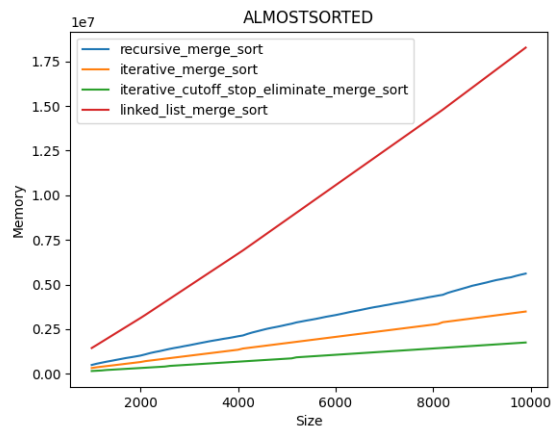
Кількість проведених порівнянь:



Операції "копіювань":

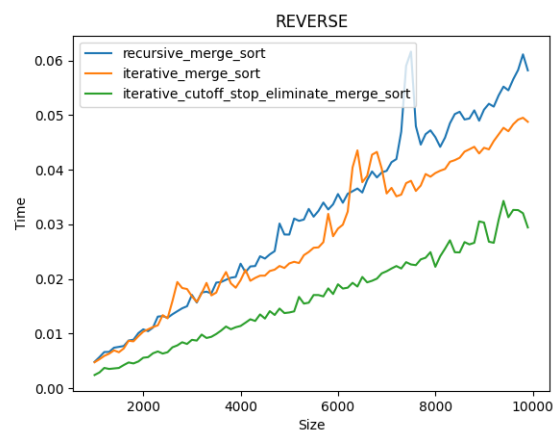
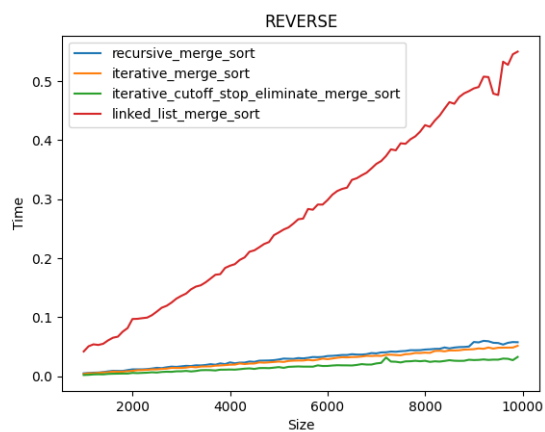


Використано пам'яті:

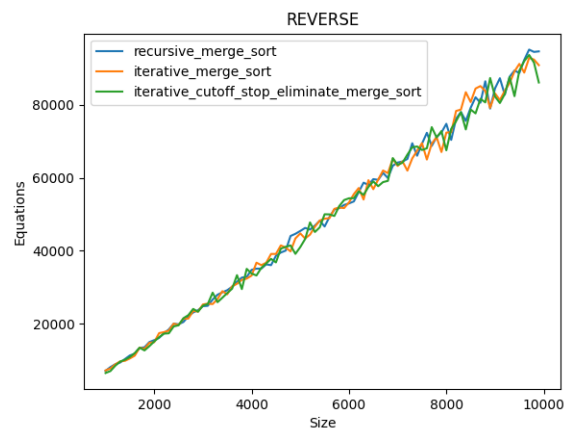
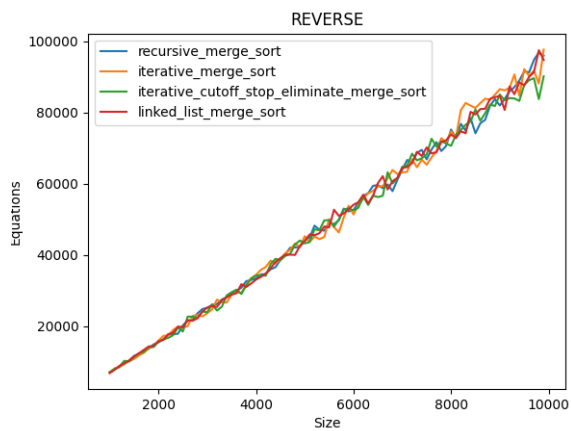




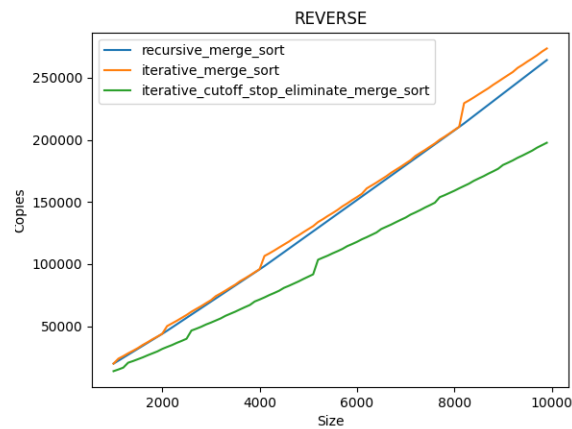
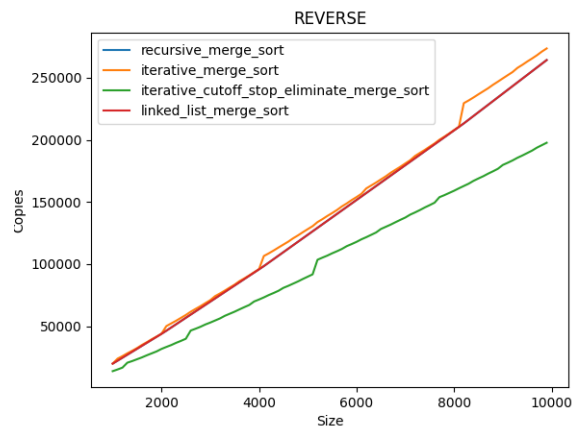
Масив відсортованих значень у зворотному порядку:  
Час виконання:



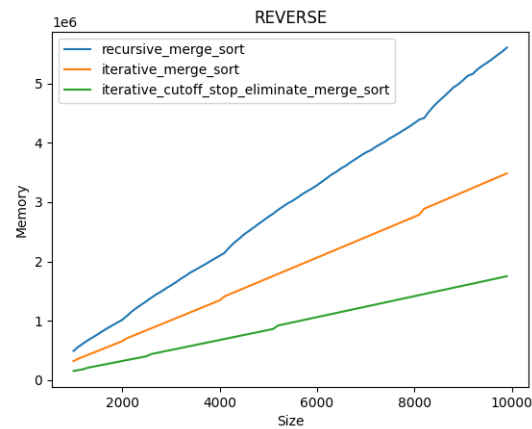
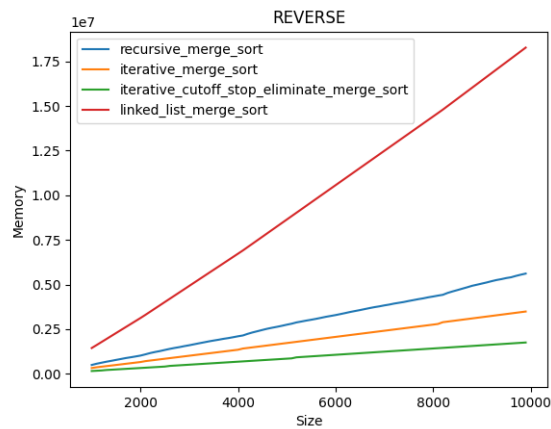
Кількість проведених порівнянь:



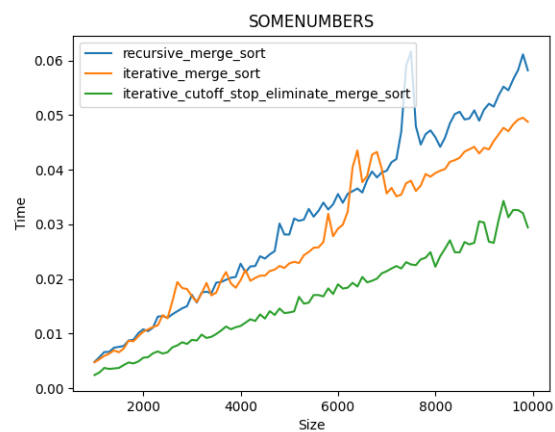
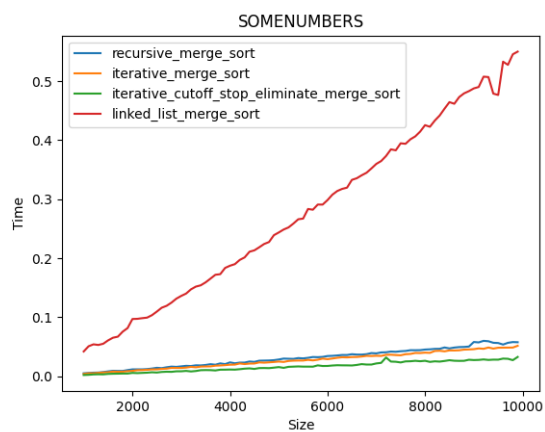
Операції "копіювань":



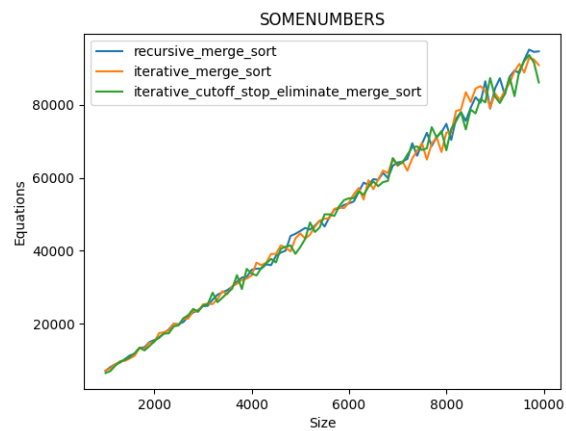
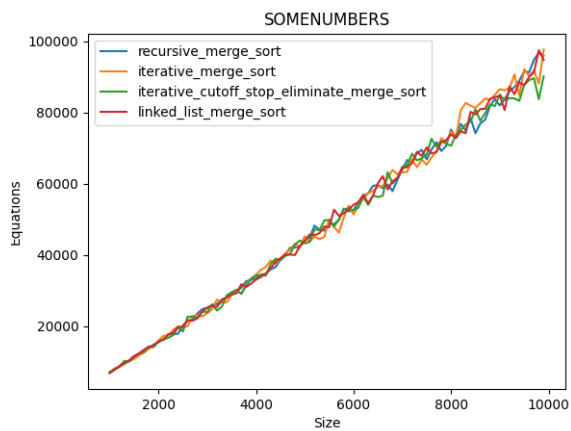
Використано пам'яті:



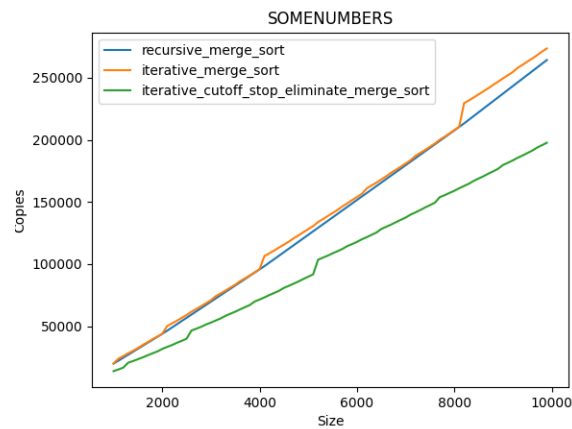
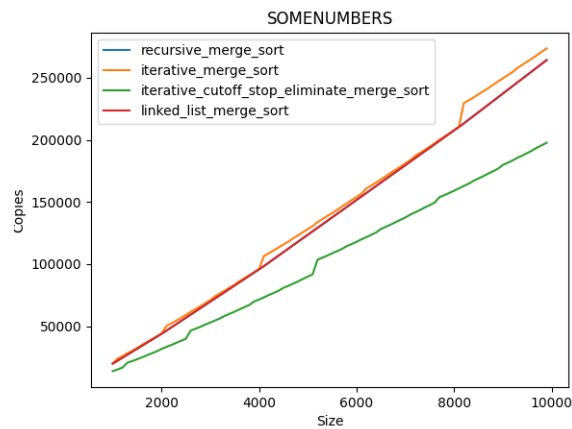
**Масив тільки з декількома різними значень:**  
Час виконання:



Кількість проведених порівнянь:



Операції "копіювань":



Використано пам'яті:

