Merge Sort

Dadmo

November 28, 2024

Contents

1 Merge Sort **1**

2 Random Lists **2**

3 Linked Lists **2**

4 Converters **2**

5 Comparisions and Results **3**
5.1 Спочатку для відсортованих масивів: 3
5.2 Потім для масивів випадкових значень: 5
5.3 Масив майже відсортованих значень: 7
5.4 Масив відсортованих значень у зворотному порядку: 9
5.5 Масив тільки з декількома різними значень: 11
5.6 Results: 13

1 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. Сотування 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 массиви розміру між $80{\cdot}10^A$ та $90{\cdot}10^A$, де A — const, сортуються не до кінця (Останні

c 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 відстотків обрахунки вже завершено.

2 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).

3 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.

4 Converters

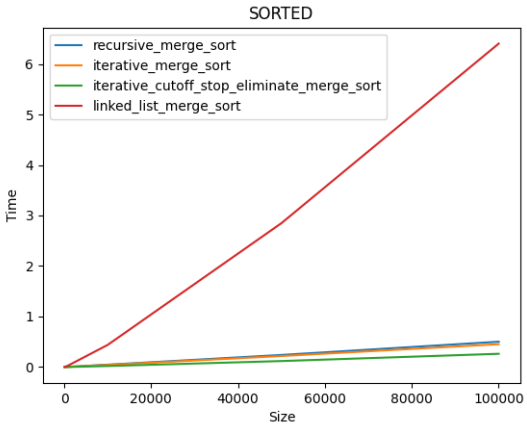
Converter — клас, в якому реалізовано конвертацію масиву у зв'язний список та навпаки.

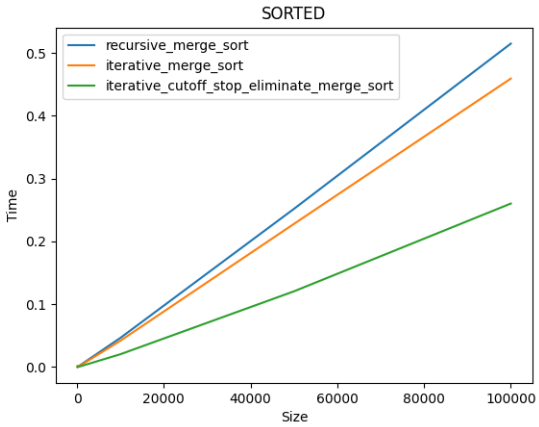
5 Comparisions and Results

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

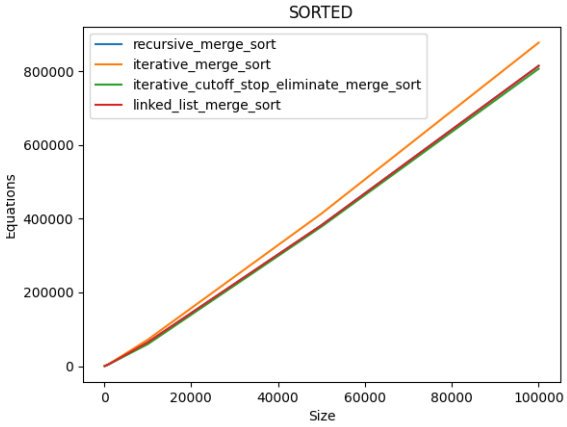
5.1 Спочатку для відсортованих масивів:

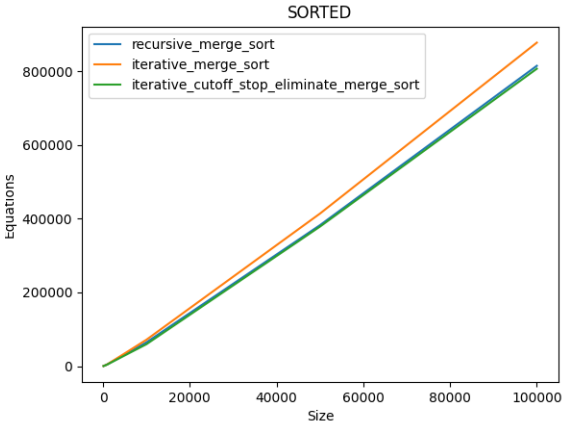
Час виконання:



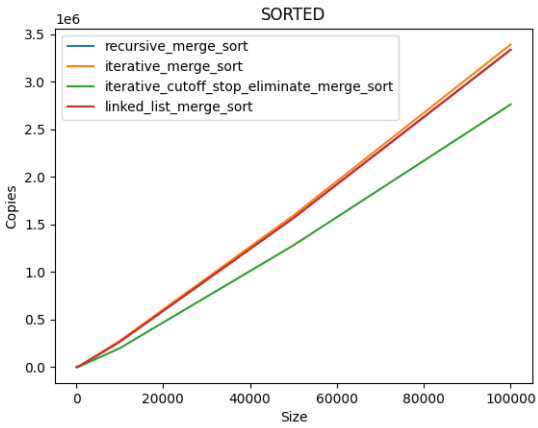


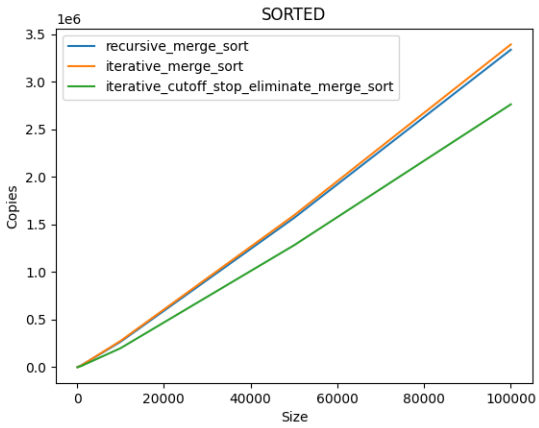
Кількість проведених порівнянь:



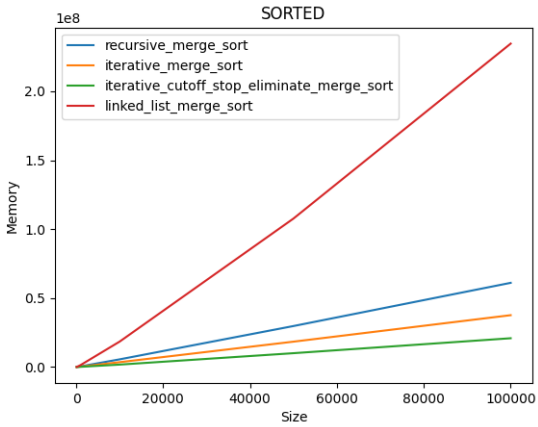


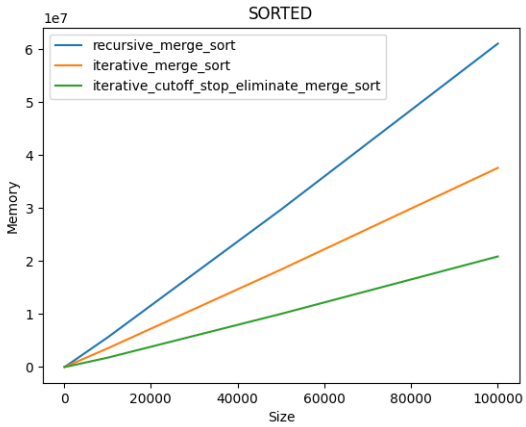
Операції "копіювань":





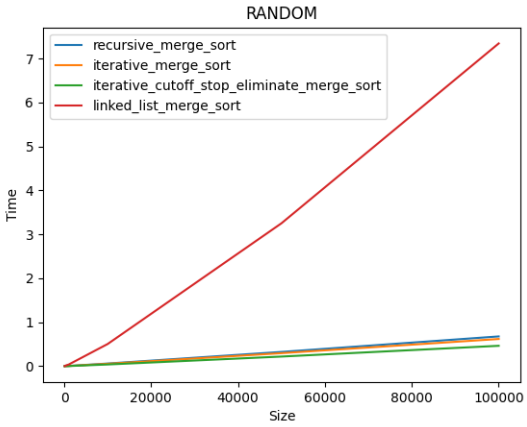
Використано пам'яті:

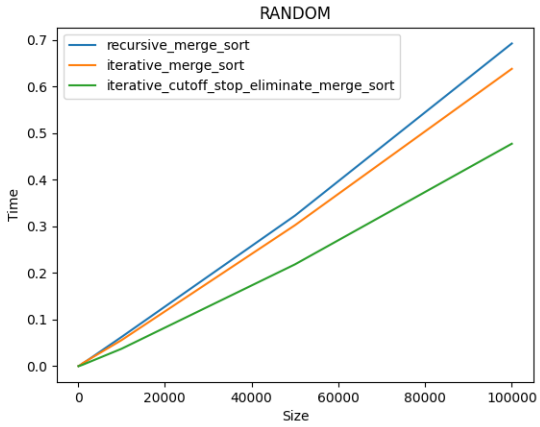




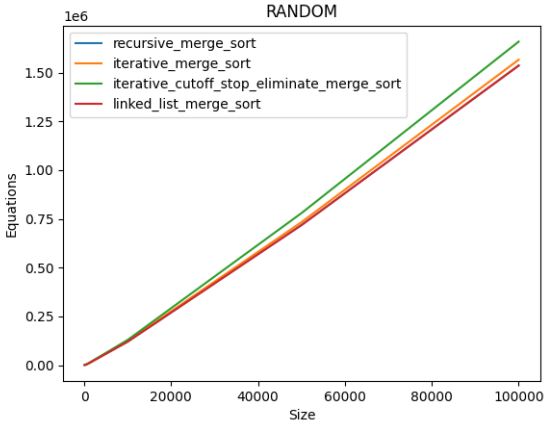
5.2 Потім для масивів випадкових значень:

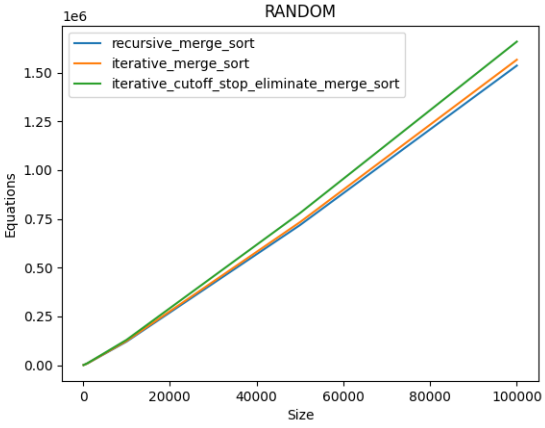
Час виконання:



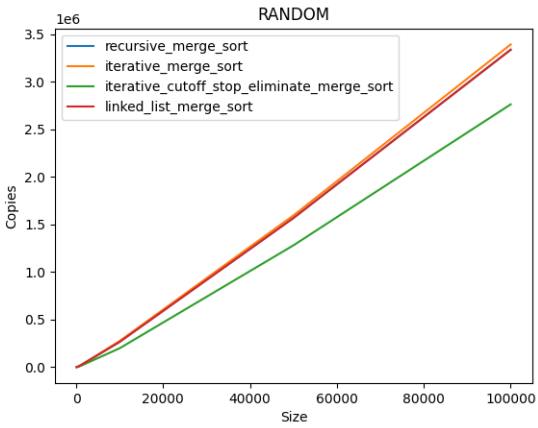


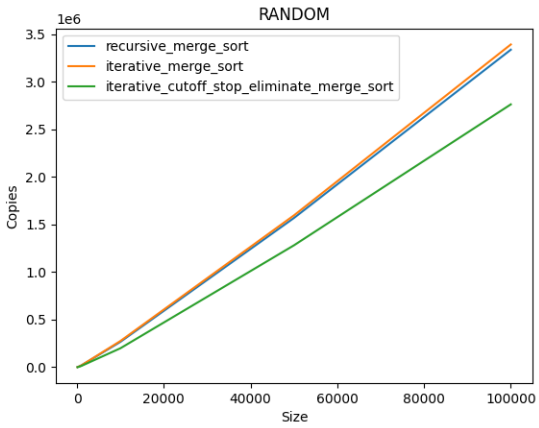
Кількість проведених порівнянь:



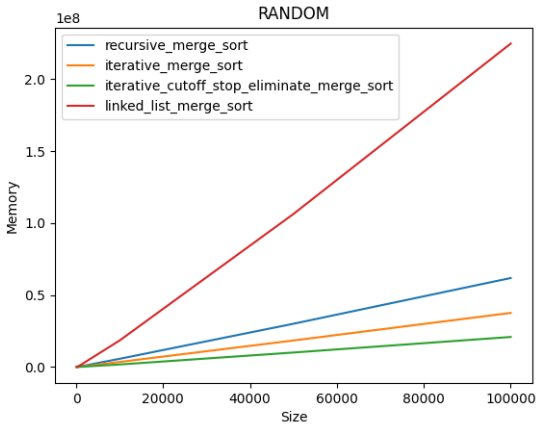


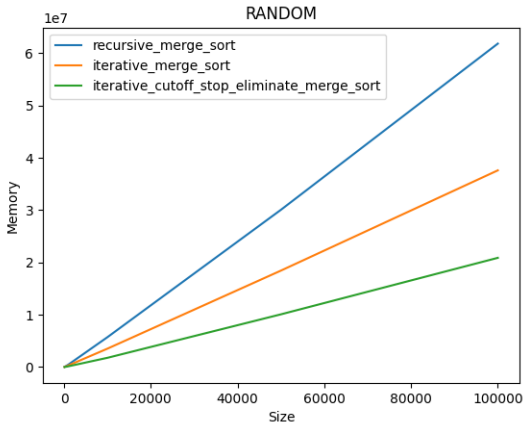
Операції "копіювань":





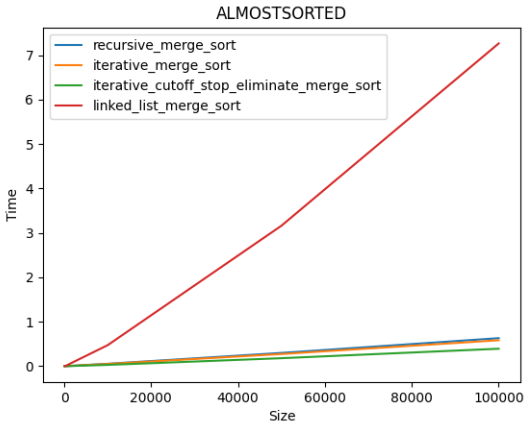
Використано пам'яті:

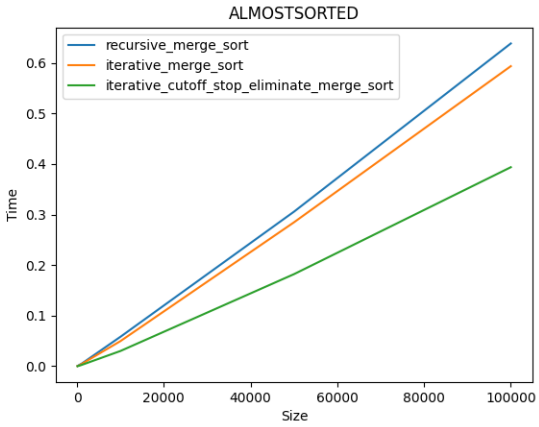




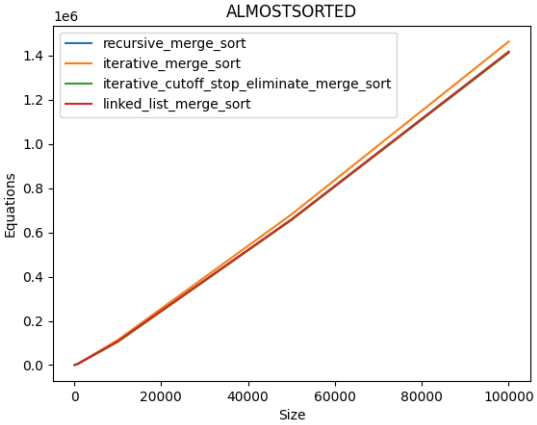
5.3 Масив майже відсортованих значень:

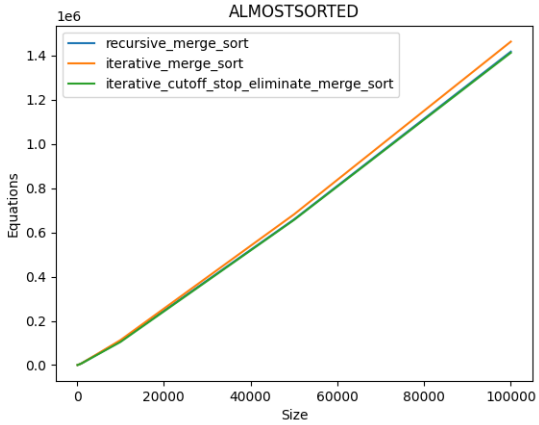
Час виконання:



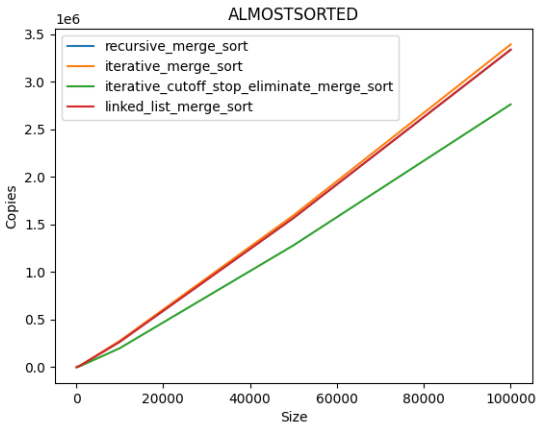


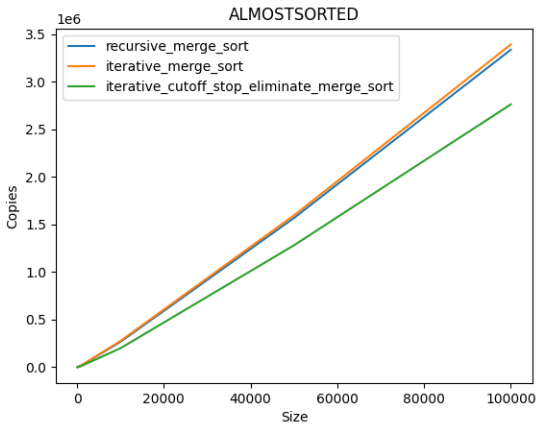
Кількість проведених порівнянь:



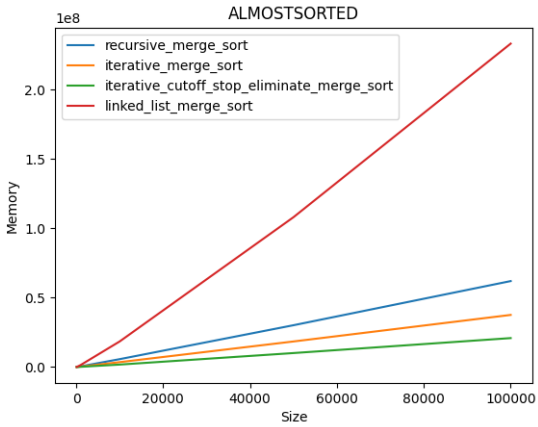


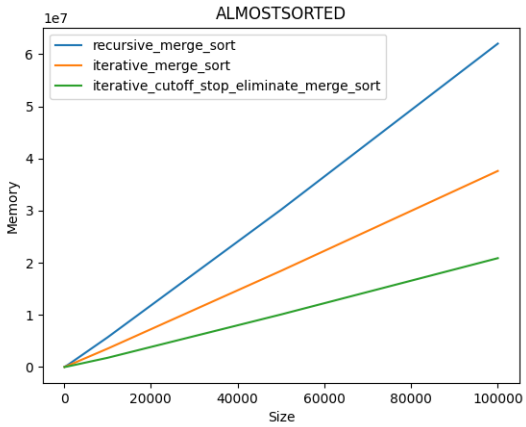
Операції "копіювань":





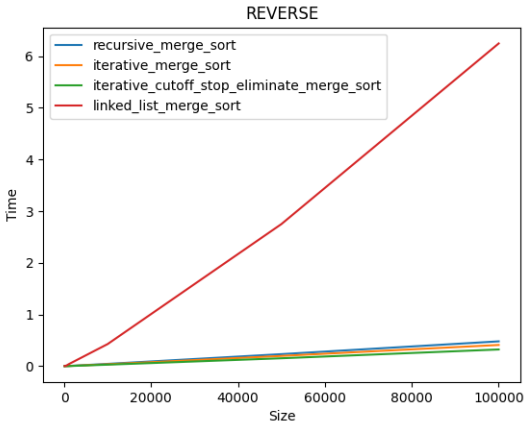
Використано пам'яті:

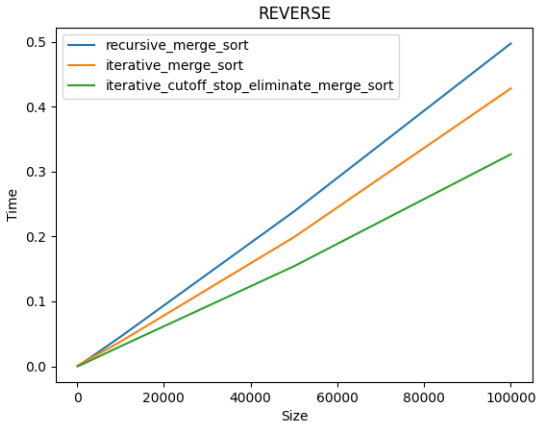




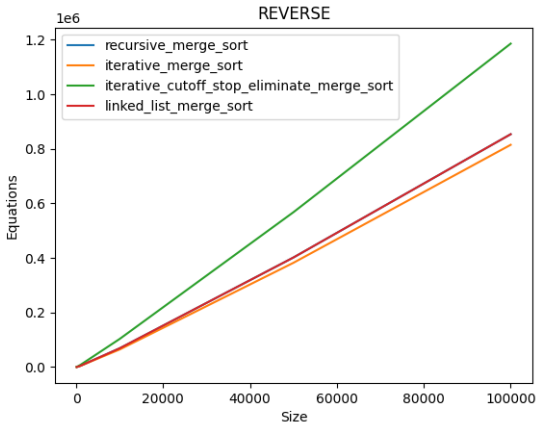
5.4 Масив відсортованих значень у зворотному порядку:

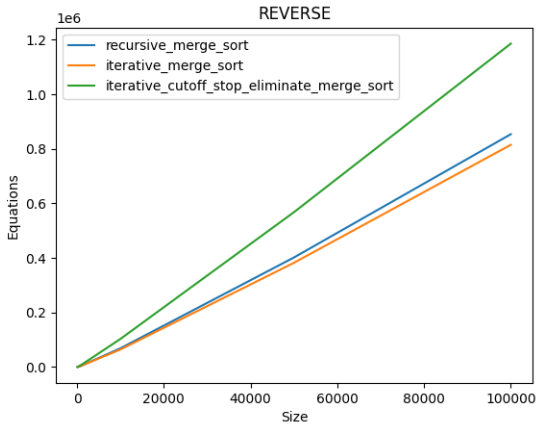
Час виконання:



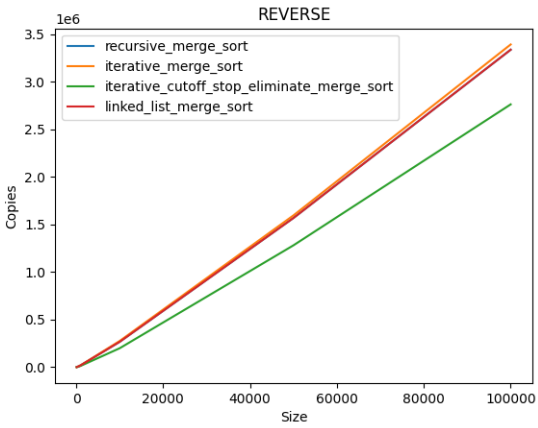


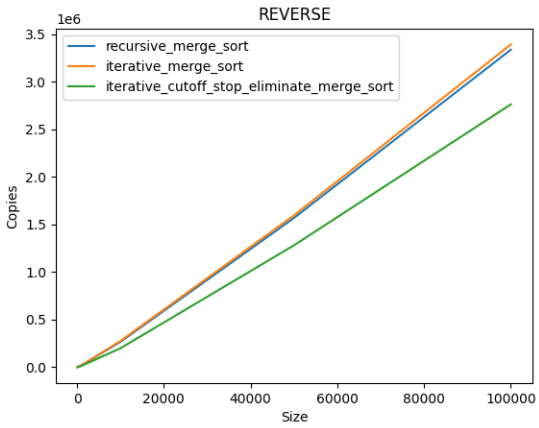
Кількість проведених порівнянь:



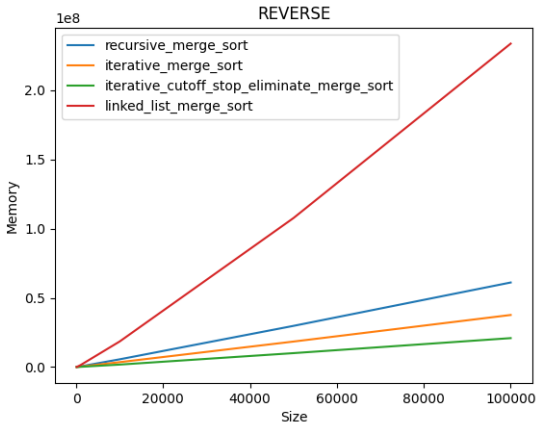


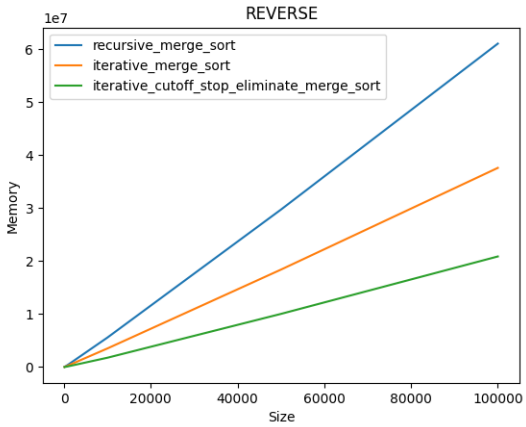
Операції "копіювань":





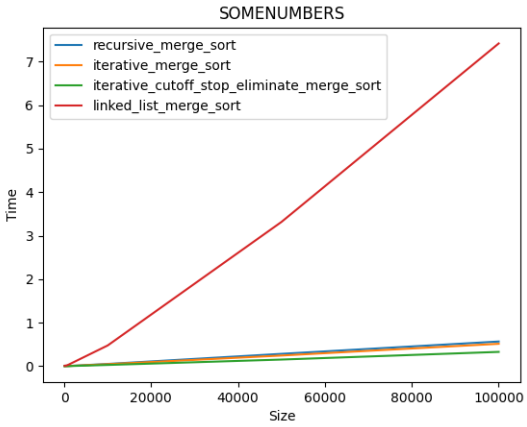
Використано пам'яті:

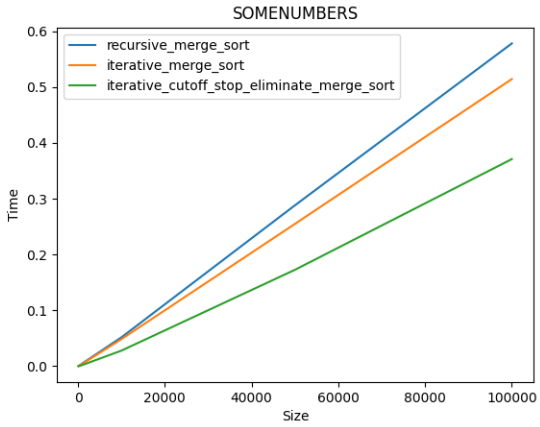




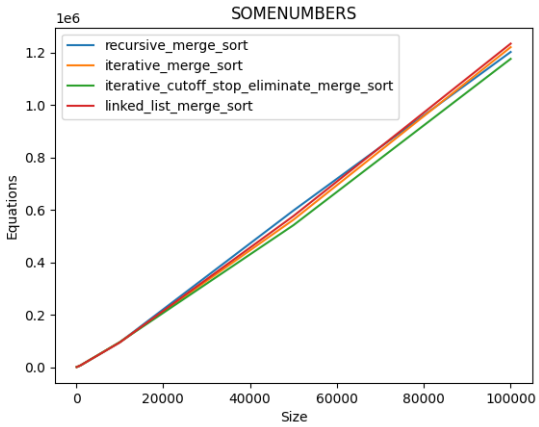
5.5 Масив тільки з декількома різними значень:

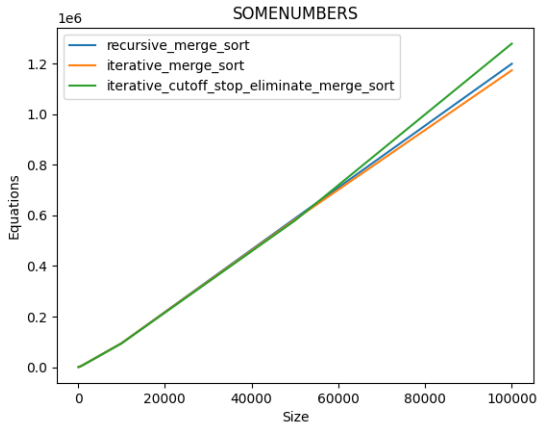
Час виконання:



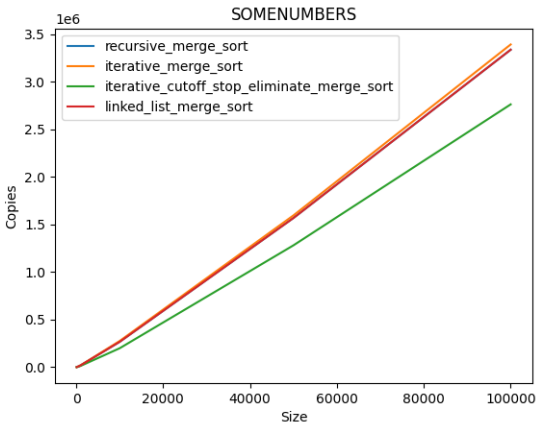


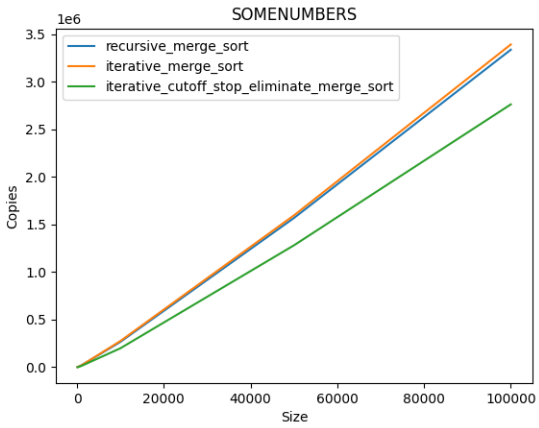
Кількість проведених порівнянь:



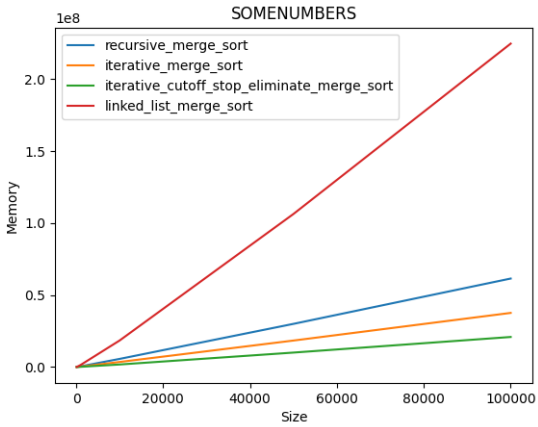


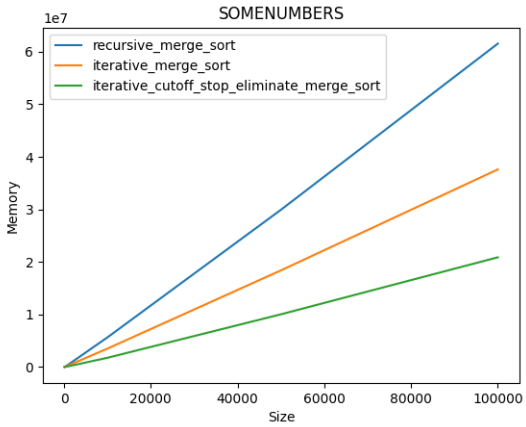
Операції "копіювань":





Використано пам'яті:





5.6 Results:

1. Час виконання — сортування злиттям для списків завжди має найбільший час виконання через те, як я їх реалізував.
2. Час виконання — Найшвидшим завжди є ітеративний з оптимізаціями.
3. Кількість проведених порівнянь — Ітеративний без оптимізацій завжди має найбільшу кількість порівнянь для відсортованих масивів та завжди має найменшу кількість порівнянь для відсортованих у зворотному порядку масивів.
4. Кількість проведених порівнянь — Ітеративний з оптимізаціями завжди має найбільшу кількість порівнянь для відсортованих у зворотному порядку масивів.
5. Операції "копіювань" — у ітеративного з оптимізаціями завжди менше всіх.
6. Пам'ять — завжди однакова.