Лабораторна робота №7
"Quick Sort"

Роботу виконав:
Климентьєв Максим
3-го курсу
групи ФІ-21

Contents

1 Quick Sort

QuickSort — клас, який має реалізовані 4 (16) варіантів алгоритму сортування.

Реалізовано 4 варіанти розбиття:

1. Розбиття Ломуто
2. Розбиття Гоара
3. Розбиття Дейкстри
4. Двоелементне Розбиття

Для кожно розбиття реалізовано 4 варіанти вибору опорного елемента:

1. Останній елемент
2. Випадковий елемент
3. Медіана першого, середнього та останього елемента
4. Медіана трьох випадкових елементів

Маю трохи не таку, як на лекції, реалізацію для розбиття Гоара та двохстороньового розбиття, оскільки опорні елементи вилучаю, а в кінці додаю на необхідне місце.

Через це розбиття Гоара робить більше порівнянь, але менше операцій обміну.

2 Quick Sort Test

Перевіряє чи масив відсортований завдяки певній варіації алгоритму чи ні.

3 Random Lists

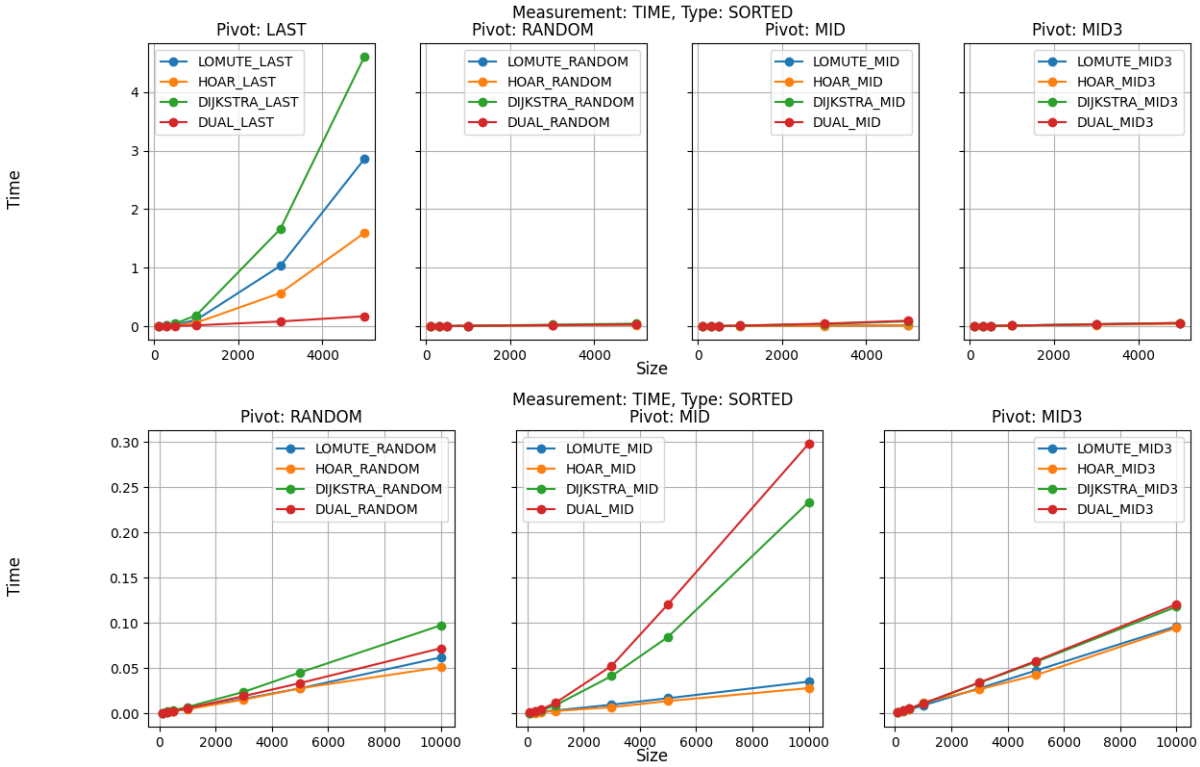
RandomLists — клас, який має реалізовані 6 варіантів генерації списків.

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).
6. **"Трикутні" (triangular)** (перша половина є строго висхідною послідовністю, а друга половина є дзеркальним відображенням першої).

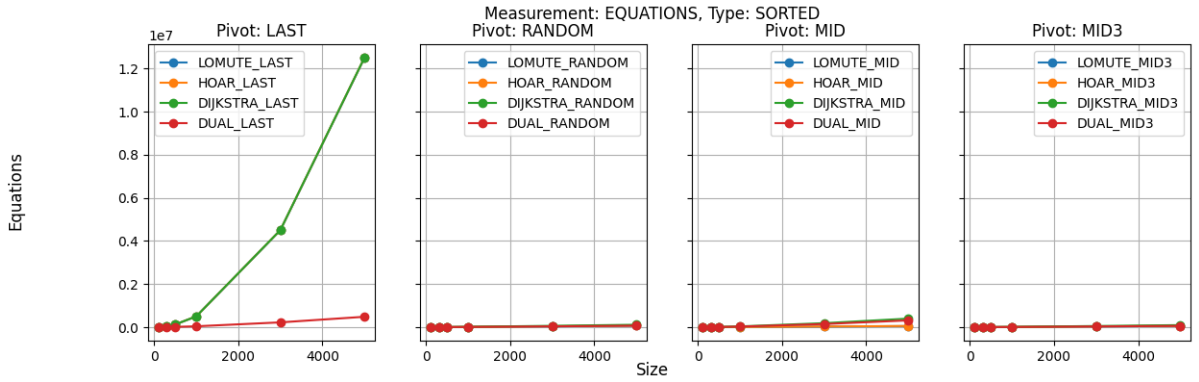
4 Comparisions and Results

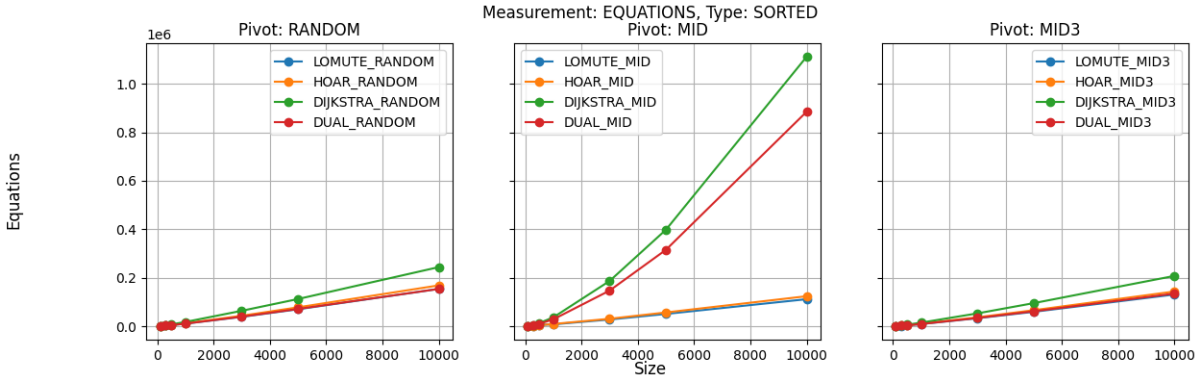
4.1 Результати для відсортованих масивів:

Час виконання:

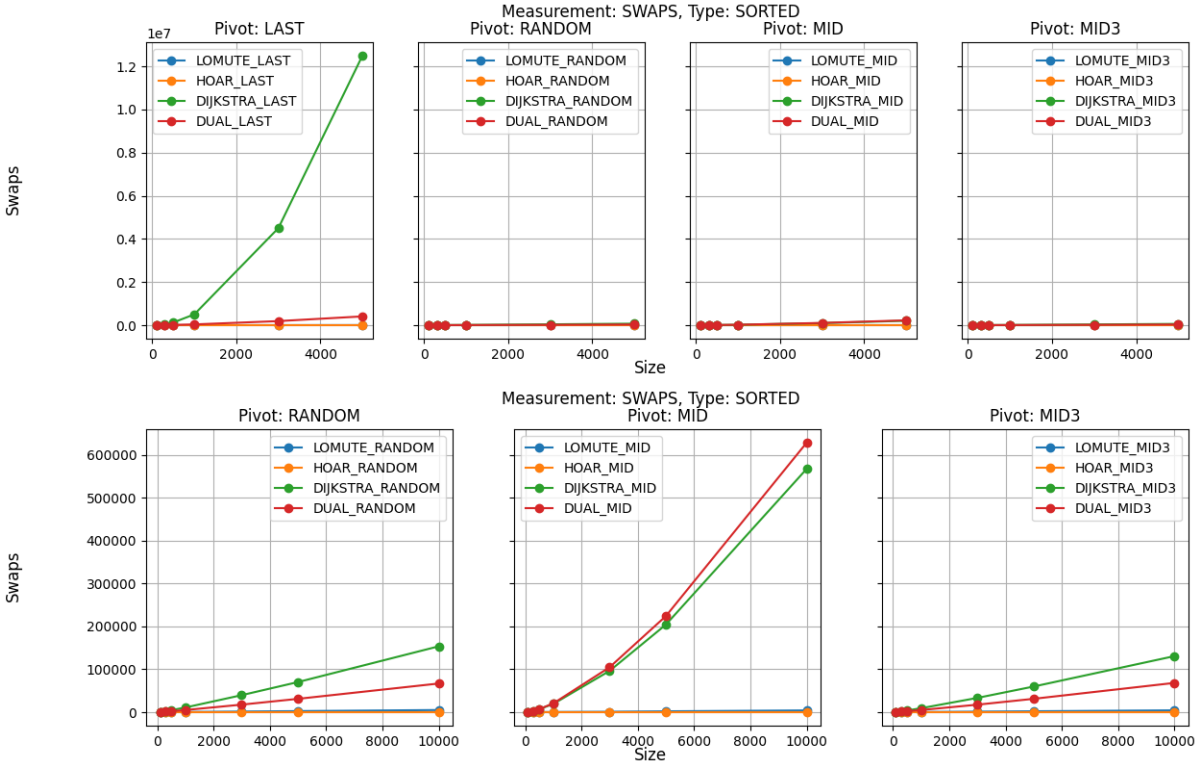


Кількість проведених порівнянь:

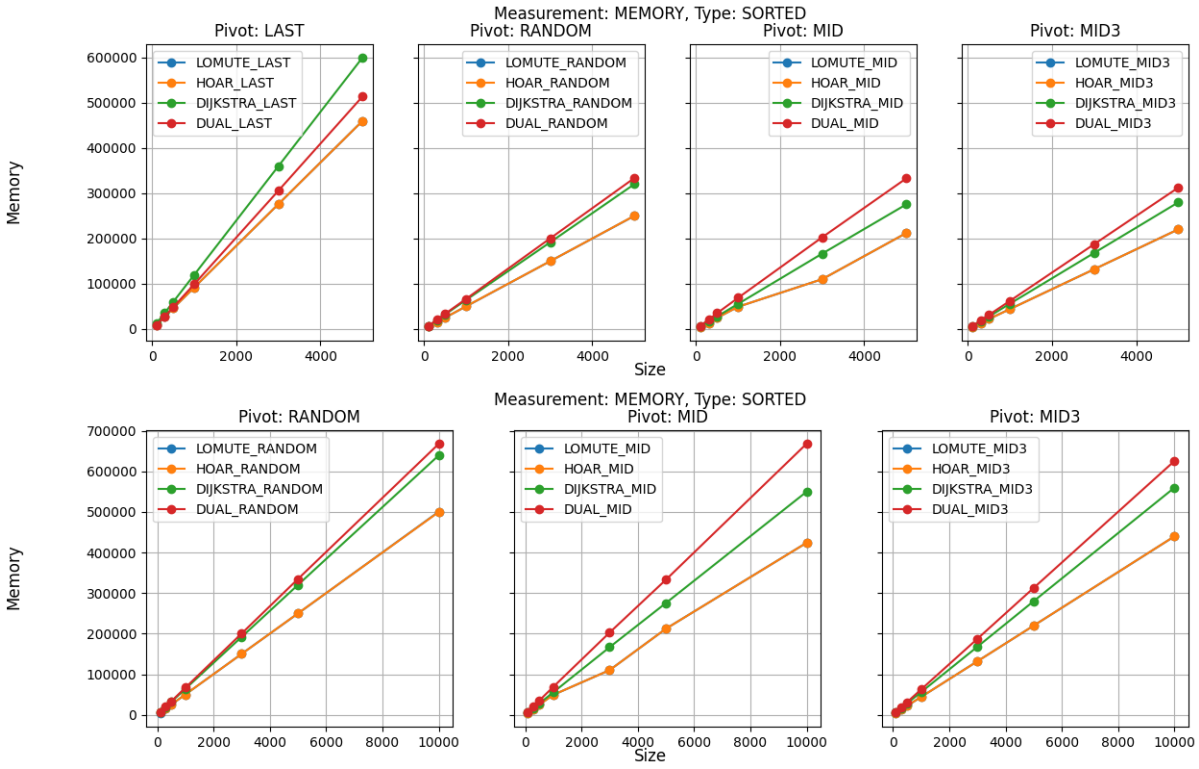




Операцій обмінів:



Використано пам'яті:



4.2 Висновки для відсортованих масивів:

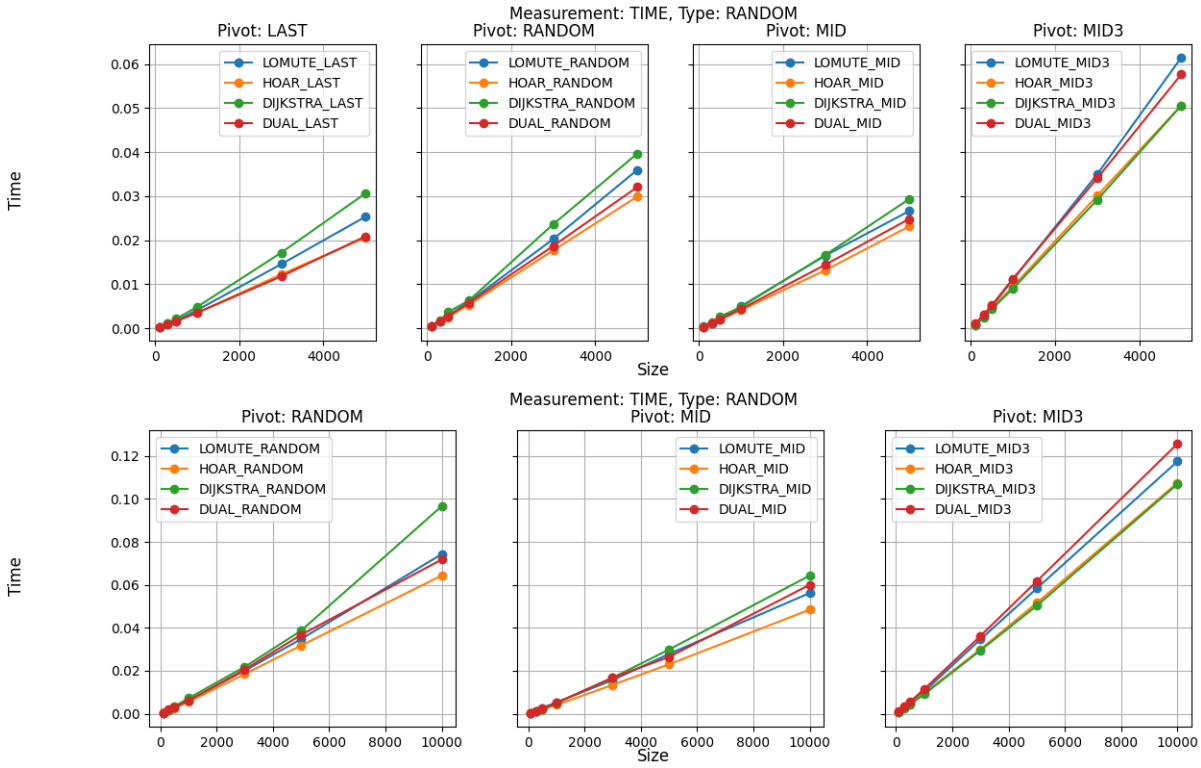
1. Усі розбиття дуже довго рахують коли опорний елемент обирається як останній.

2. Розбиття Дейкстри дуже багато порівнянь та обмінів використовує коли опорний елемент обирається як останній.

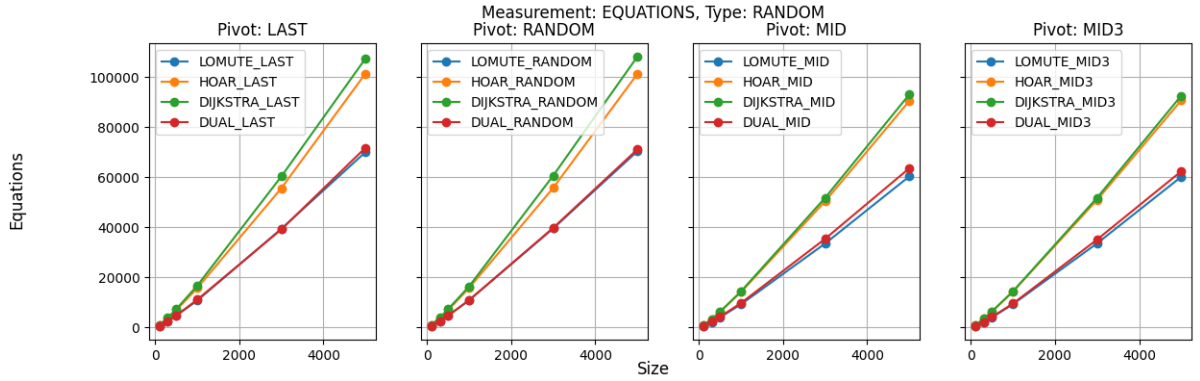
3. Найбільше пам'яті використовується коли обирається останній елемент як опорний.

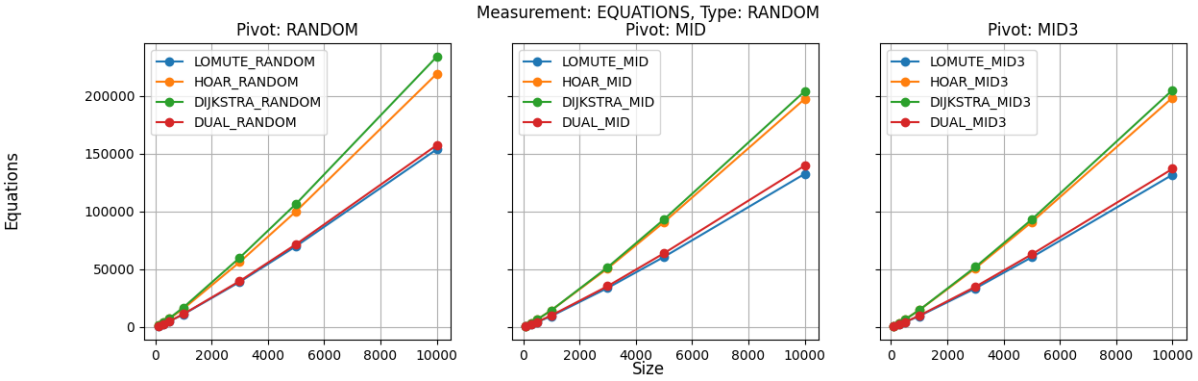
4.3 Результати для масивів випадкових елементів:

Час виконання:

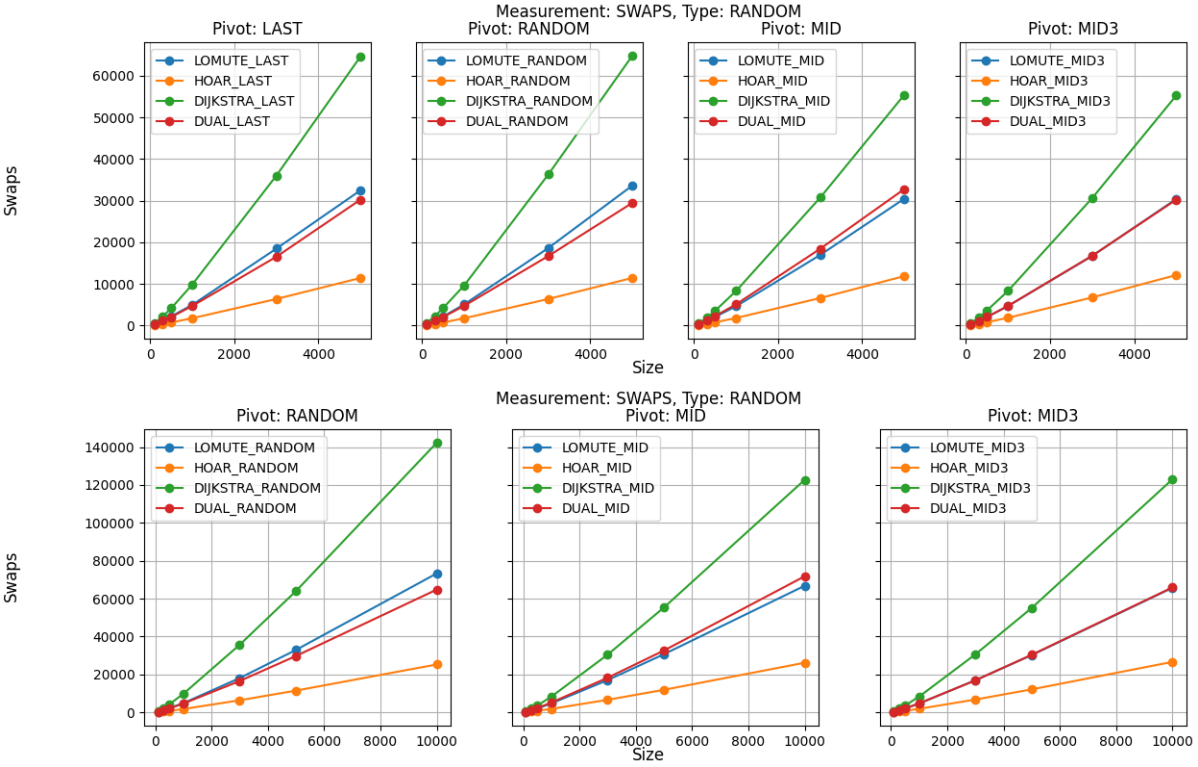


Кількість проведених порівнянь:

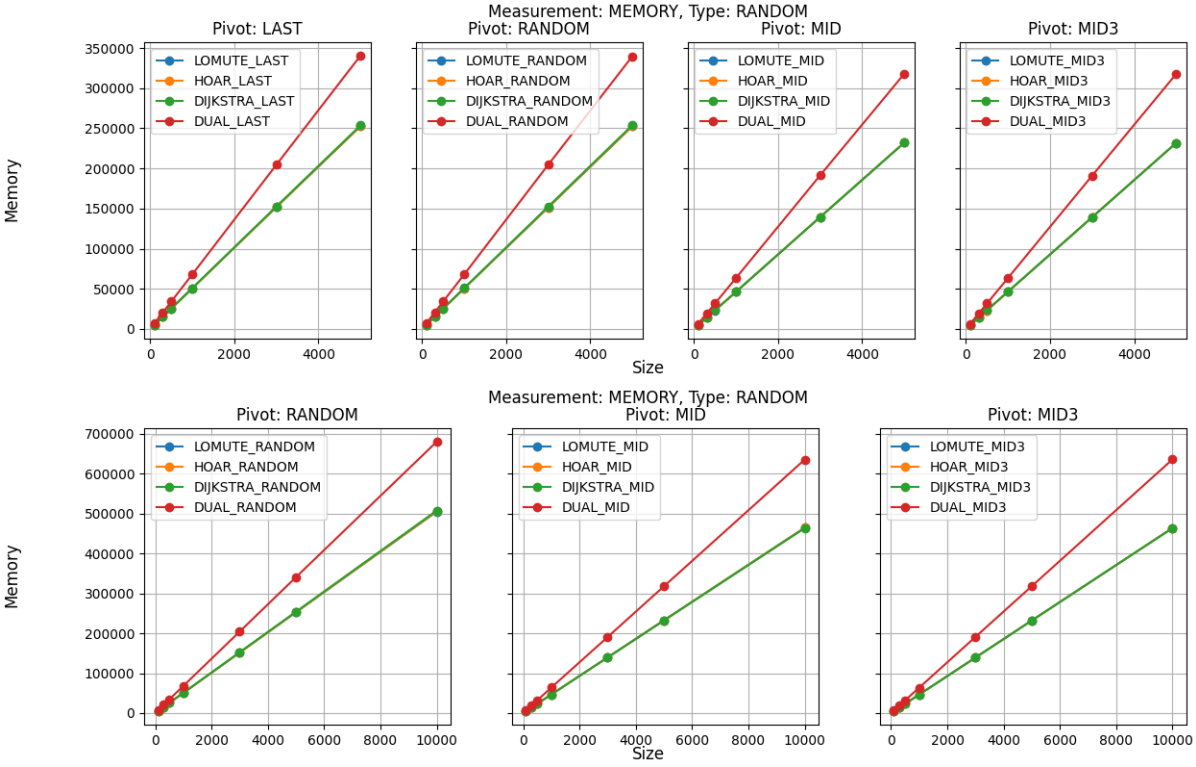




Операцій обмінів:



Використано пам'яті:



4.4 Висновки для масивів випадкових елементів:

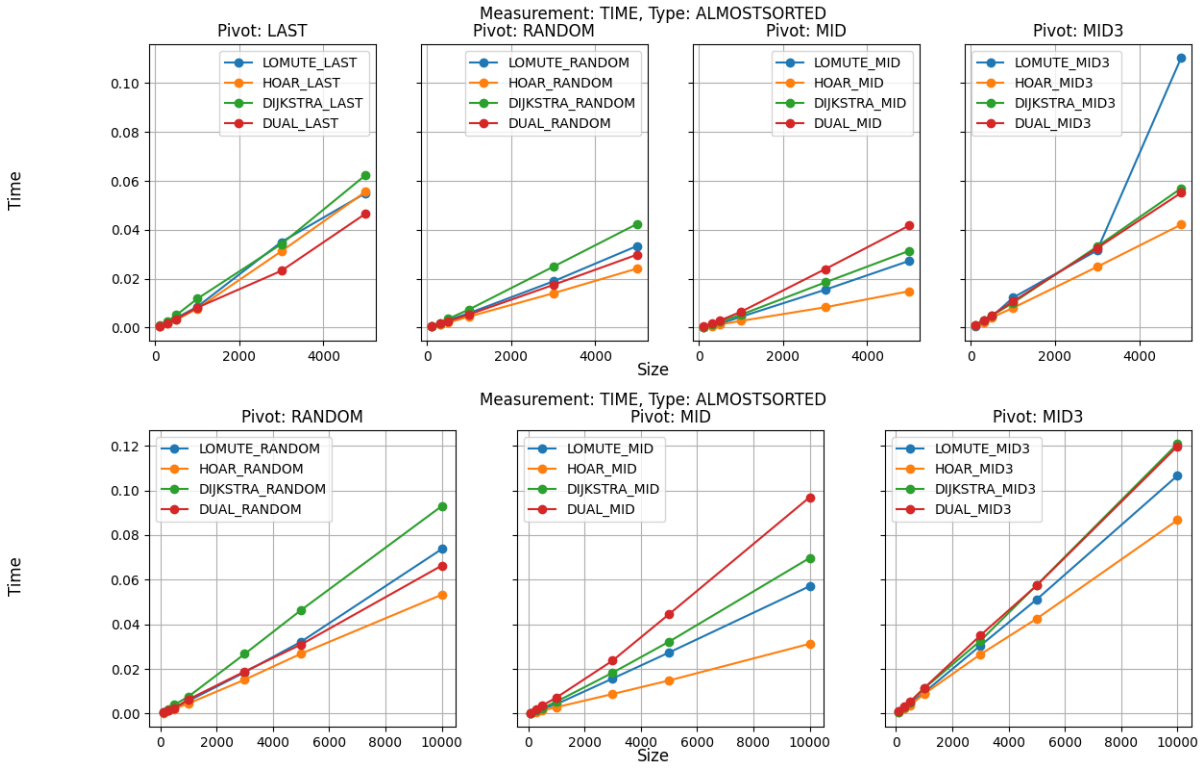
1. Для масивів з випадковими елементами обрання опорного елемента як медіани трьох випадкових елементів дуже багато часу використовує.

2. обрання опорного елементу як медіани першого, середнього та останього елемента використовує порівняно з іншим менше порівнянь, обмінів та пам'яті.

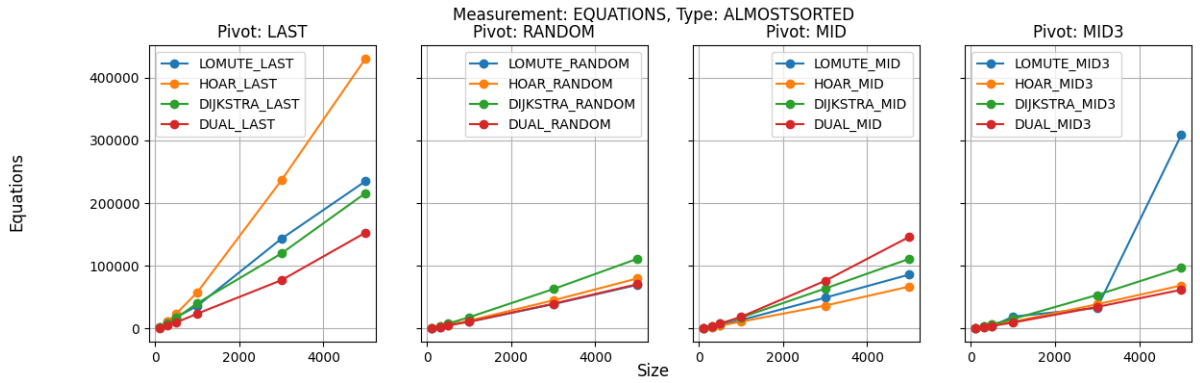
3. Дейкстра вийшла майже завжди найгірша по часу, кількості порівнянь та обмінів, але краща по пам'яті.

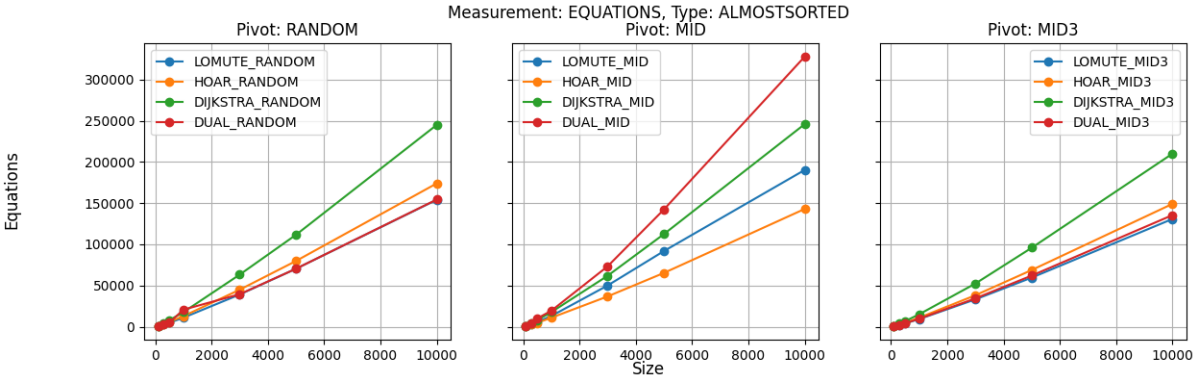
4.5 Результати для майже відсортованих масивів:

Час виконання:

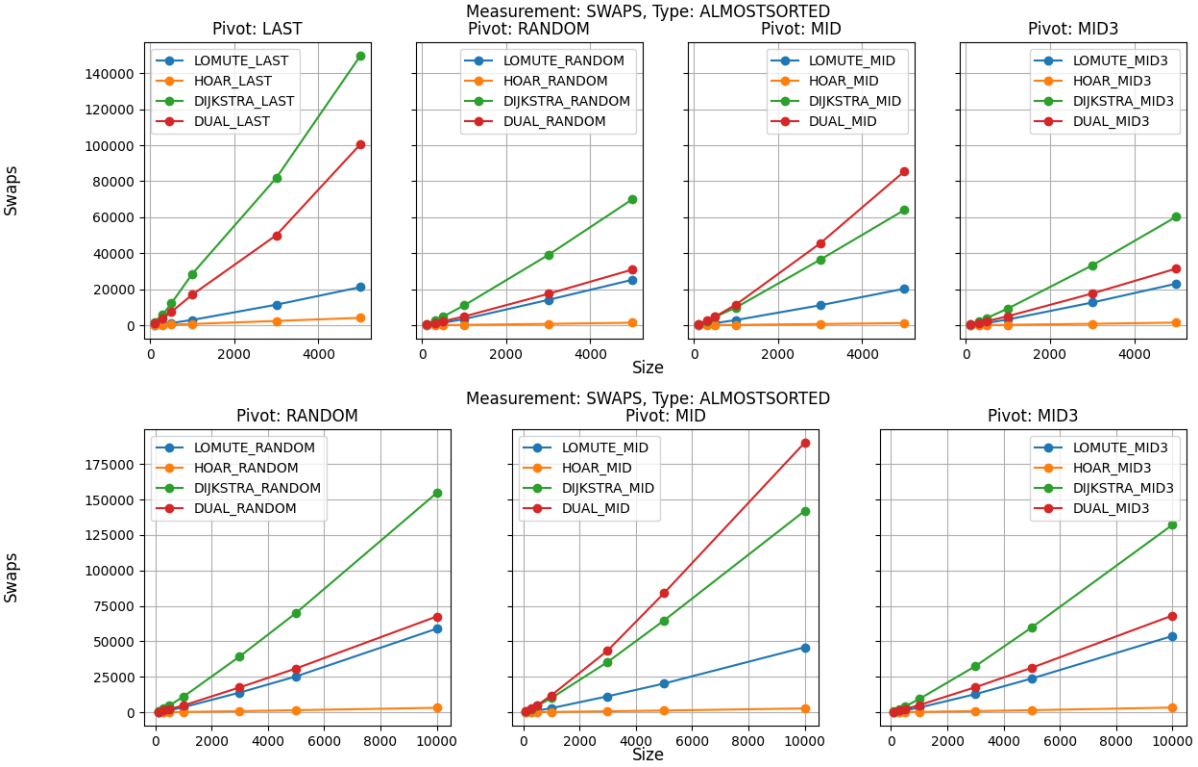


Кількість проведених порівнянь:

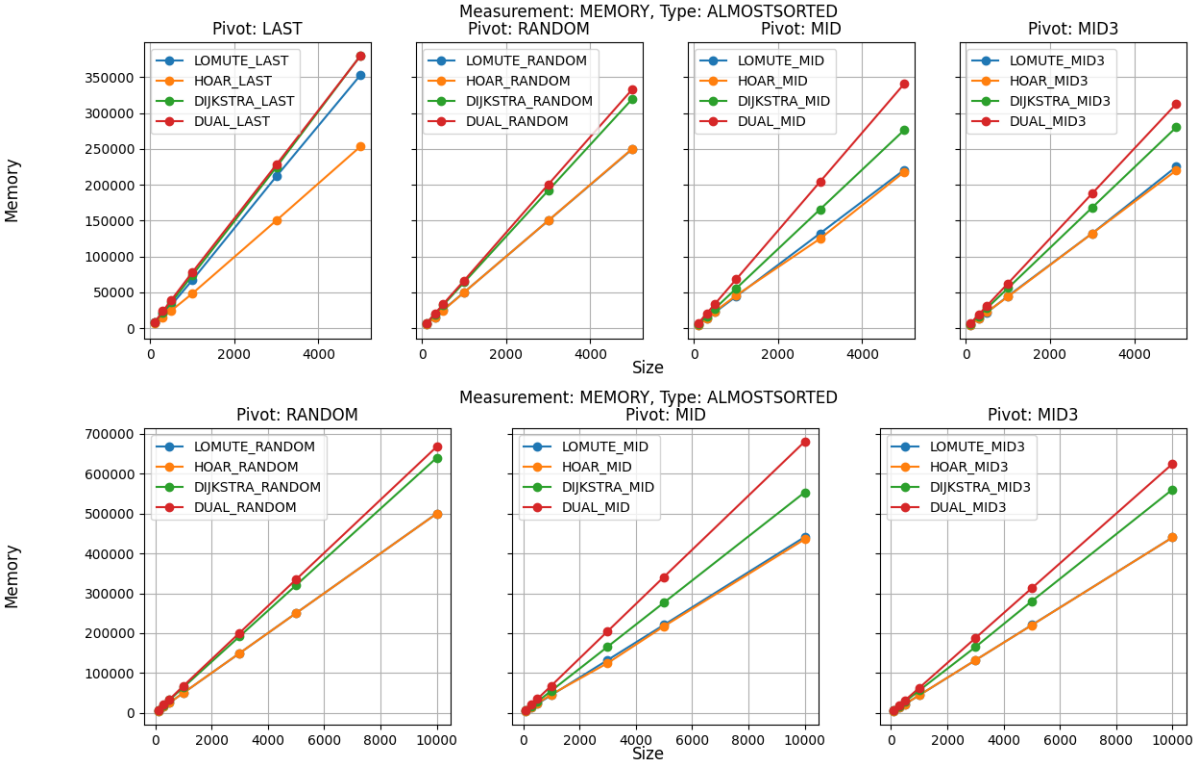




Операцій обмінів:



Використано пам'яті:



4.6 Висновки для майже відсортованих масивів:

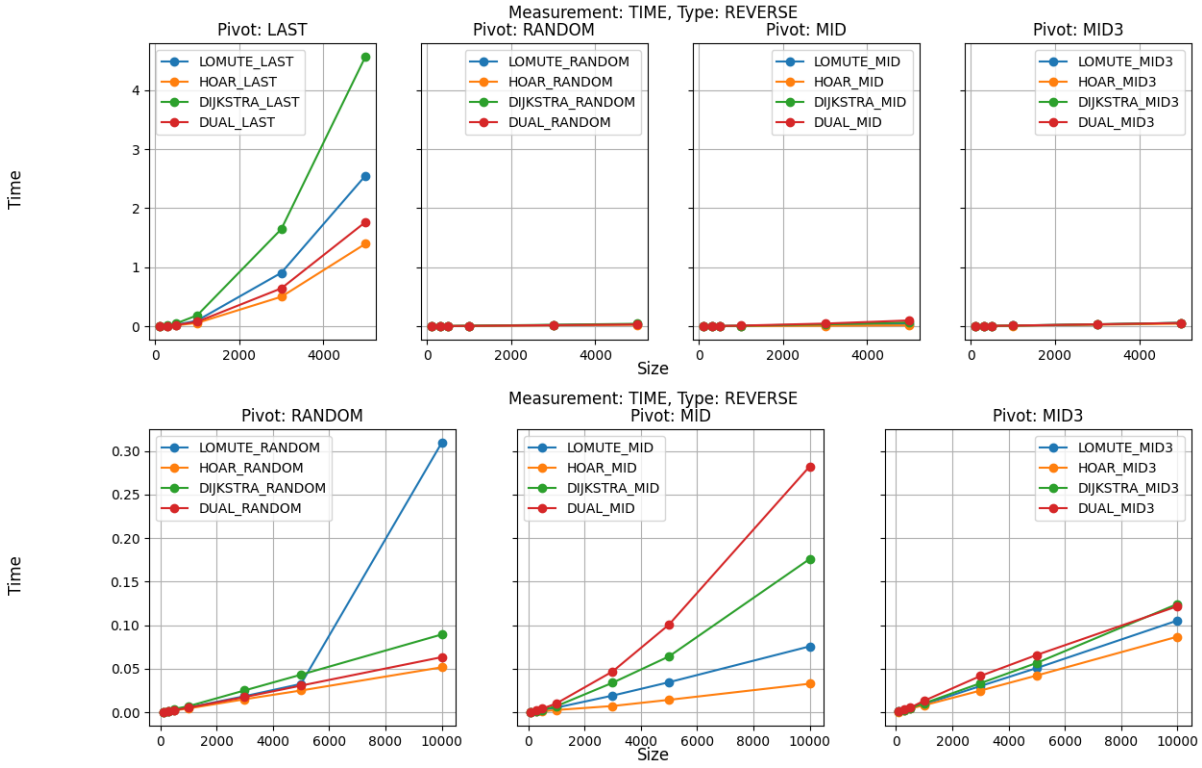
1. Для майже відсортованих масивів обрання опорного елемента як медіани трьох випадкових елементів багато часу використовує, особливо для розбиття Ломуто.

2. Гоар робить багато порівнянь, коли опорний елемент обирається як останній, а Ломуто коли як медіана трьох випадкових елементів.

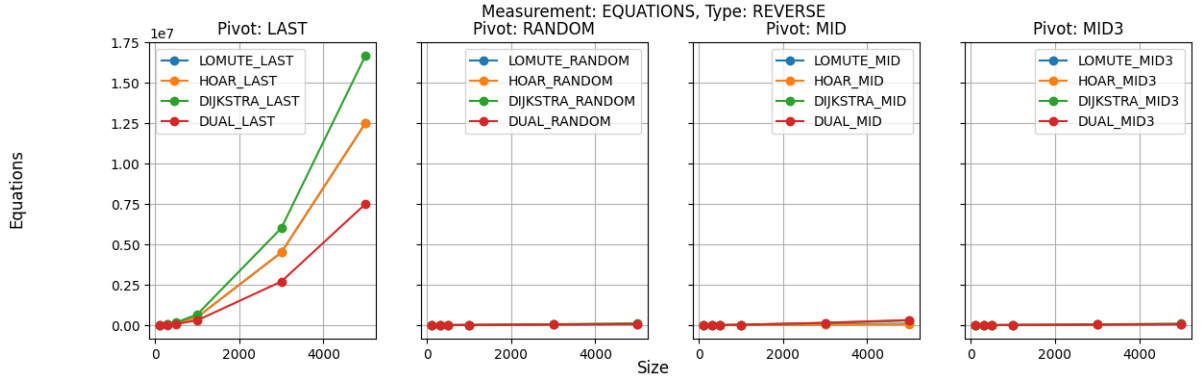
3. Дейкстра робить багата обмінів, коли опорний елементи обирається як останній.

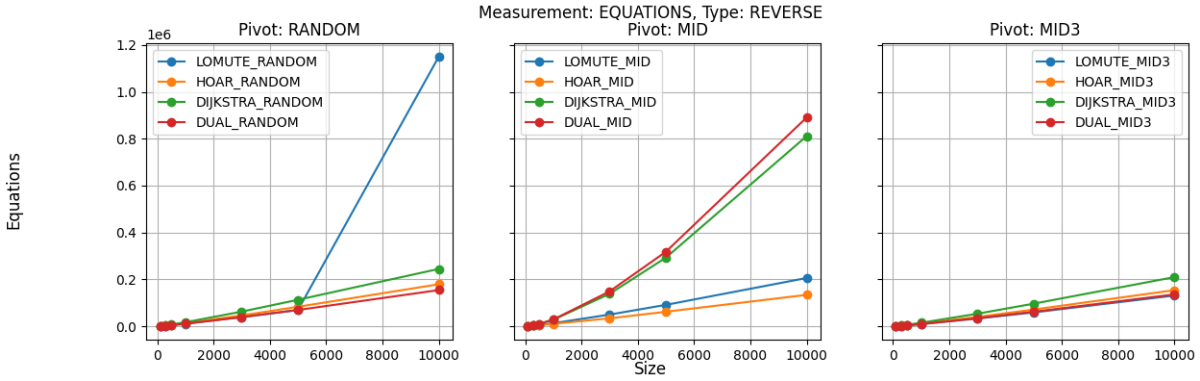
4.7 Результати для відсортованих у зворотному напрямку масивів:

Час виконання:

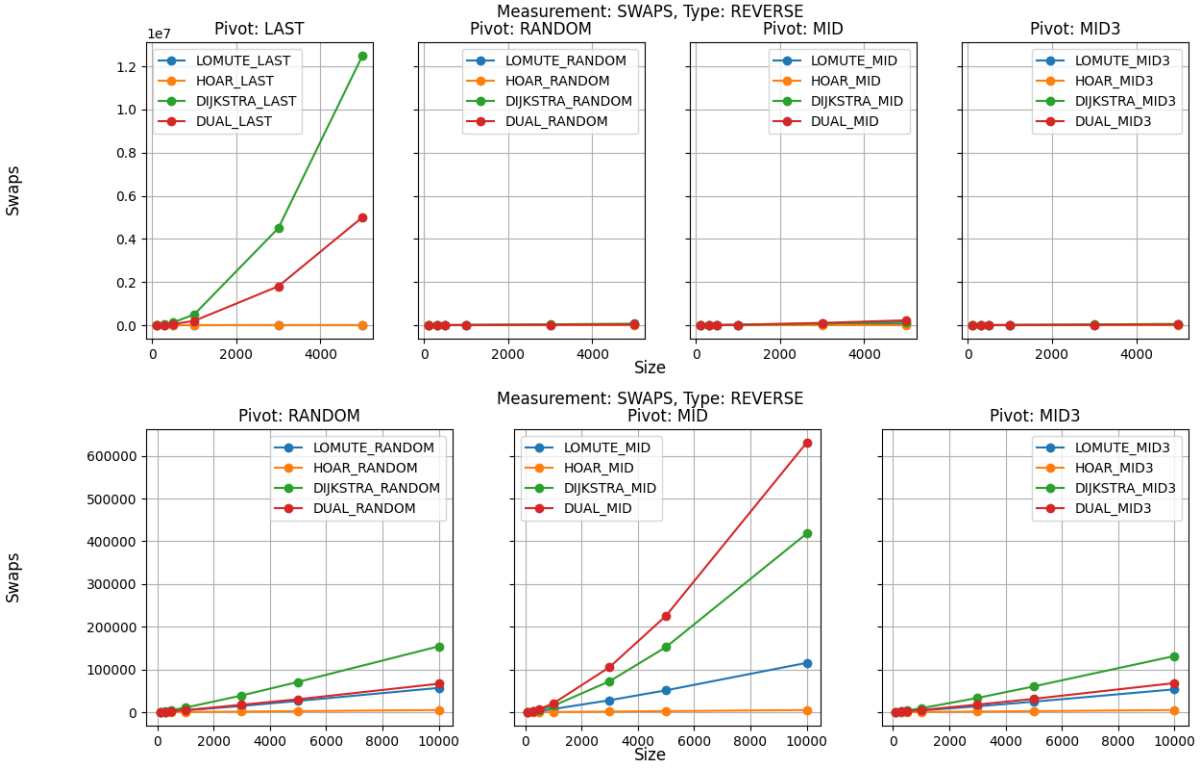


Кількість проведених порівнянь:

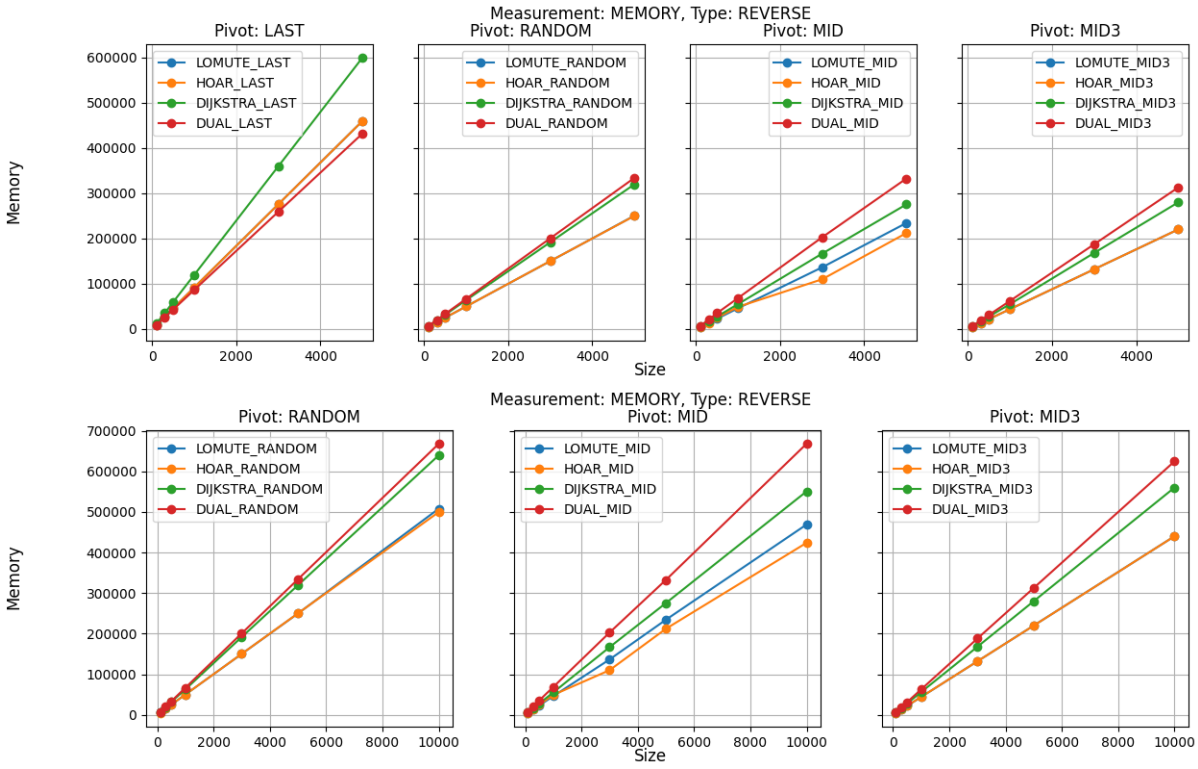




Операцій обмінів:



Використано пам'яті:



4.8 Висновки для відсортованих у зворотному напрямку масивів:

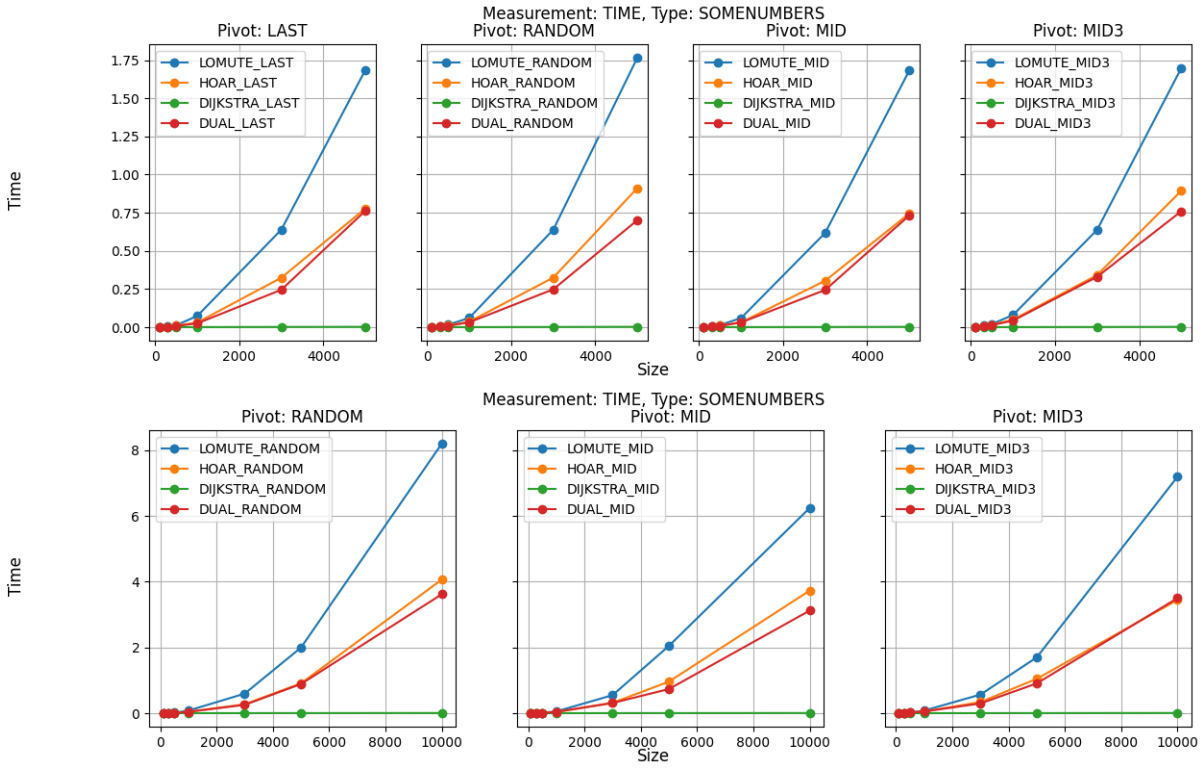
1. Дуже багато часу та порівнянь використовується, коли опорний елемент обирається як останній, через те, що два масиви, що він отримує є дуже не збалансованими по розміру, один масив має на один елемент менше, а другий не має елементів взагалі.

2. Також операцій обмінів використовується дуже багато, коли опорний елемент обирається як останній, для розбиття Дейкстри та двоелементного розбиття.

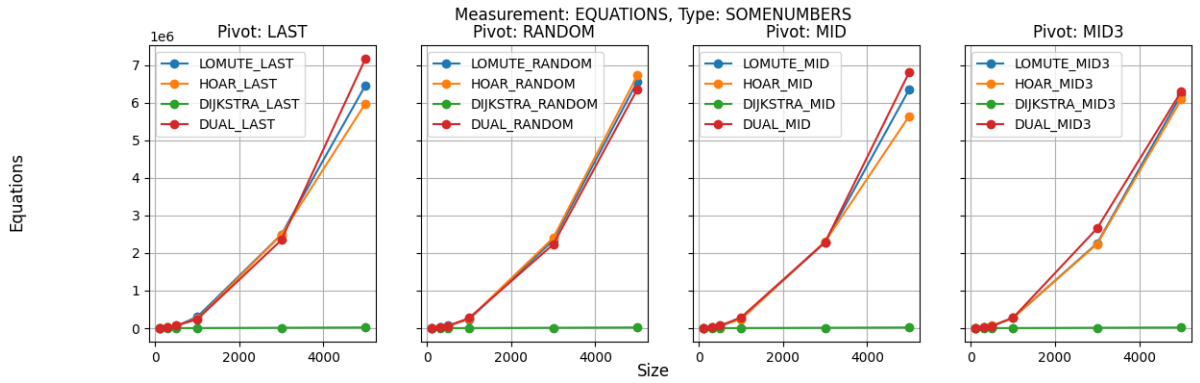
3. Гірше за все після опорного елемента, який є останнім, є медіана першого, середнього та останього елементів.

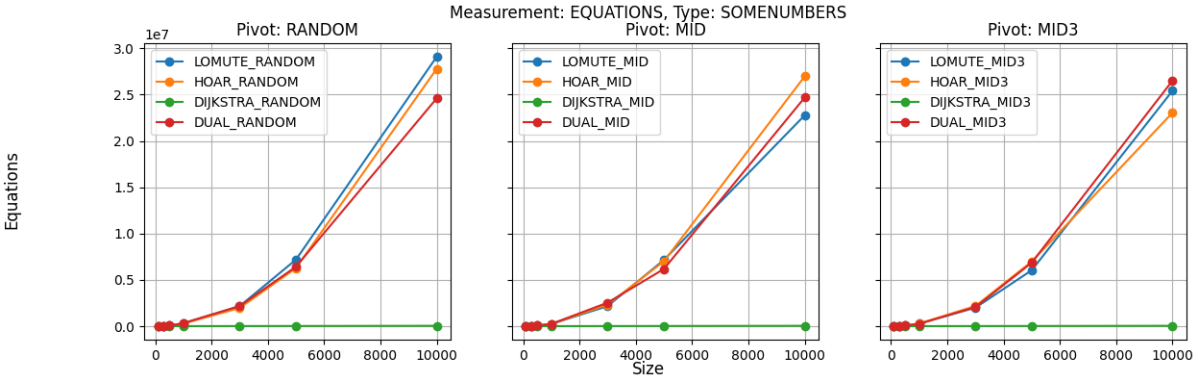
4.9 Результати для масивів лише з декількома значеннями:

Час виконання:

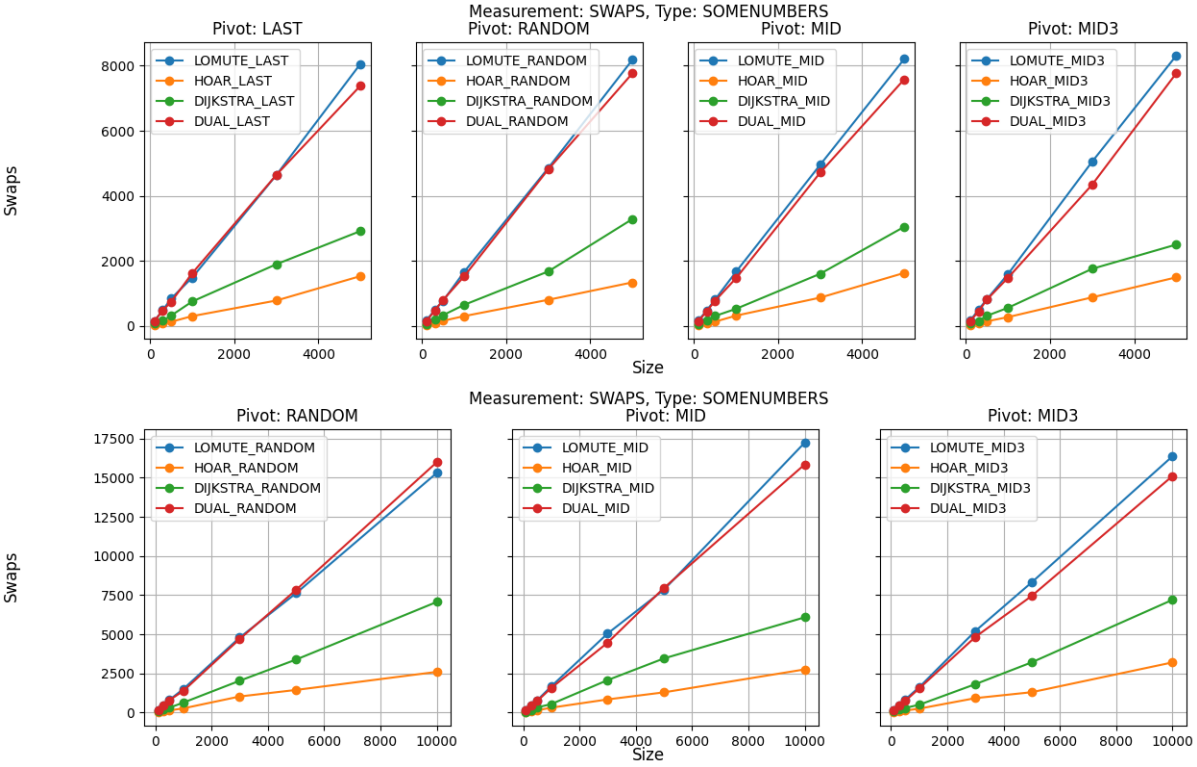


Кількість проведених порівнянь:

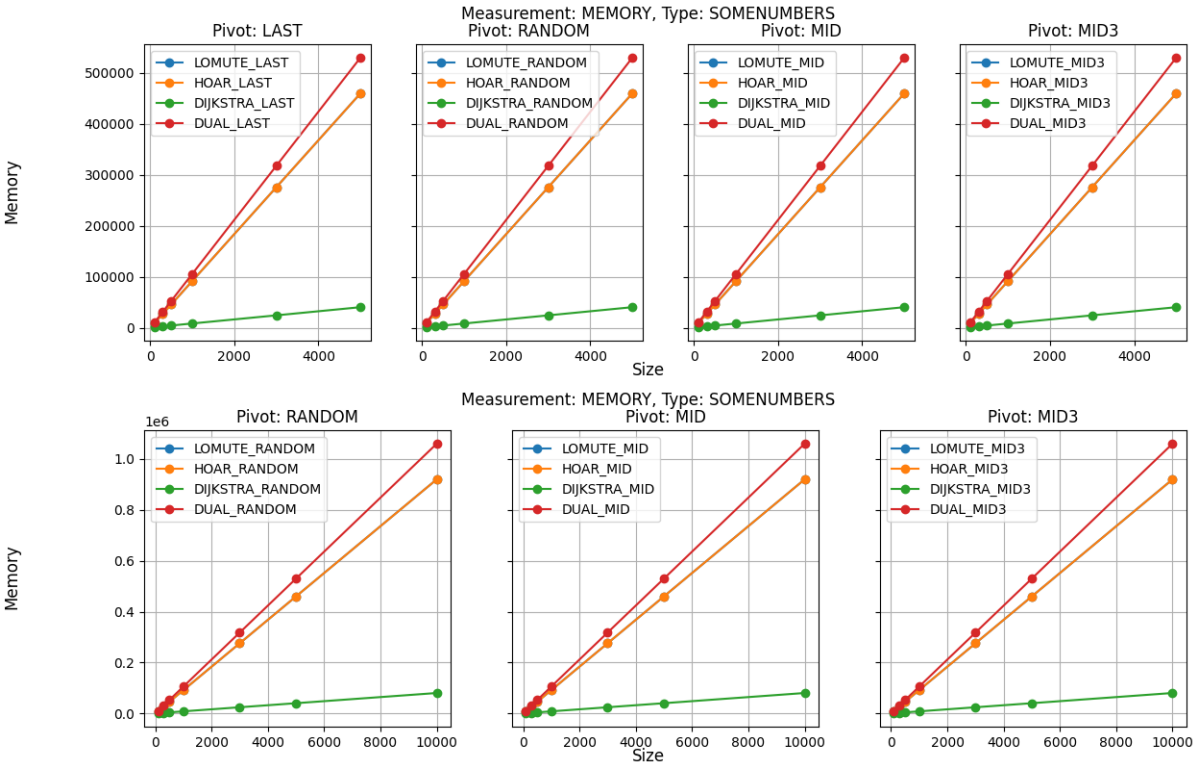




Операцій обмінів:



Використано пам'яті:

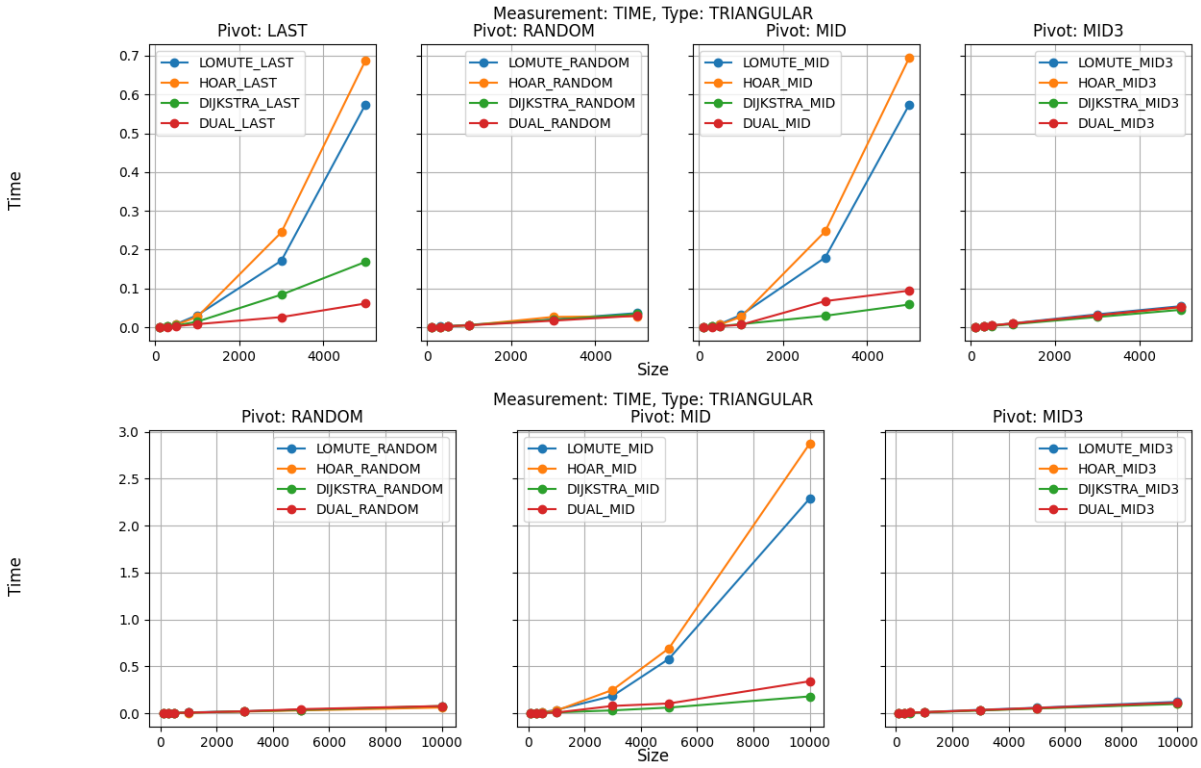


4.10 Висновки для масивів лише з декількома значеннями:

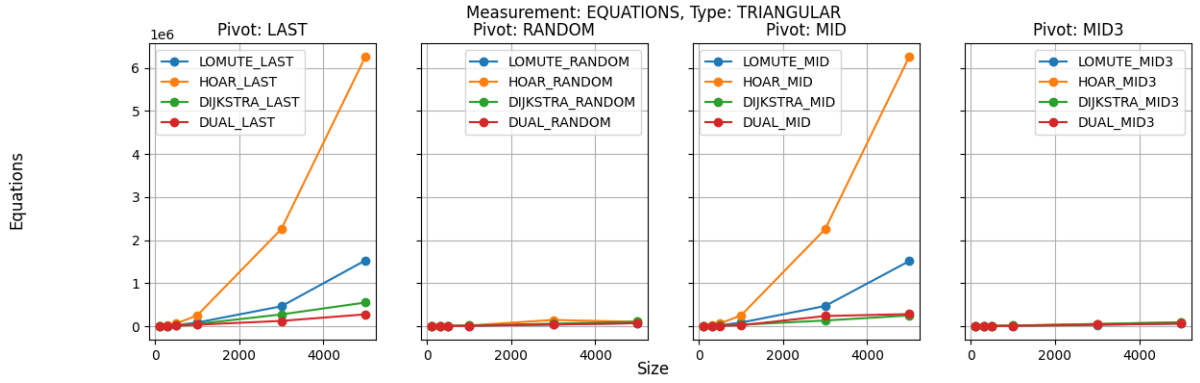
1. Дейкстра використовує найменше усього часу, порівнянь та пам'яті, а також, якщо не враховувати незвичайну реалізацію розбиття Гоара, кількість обмінів.

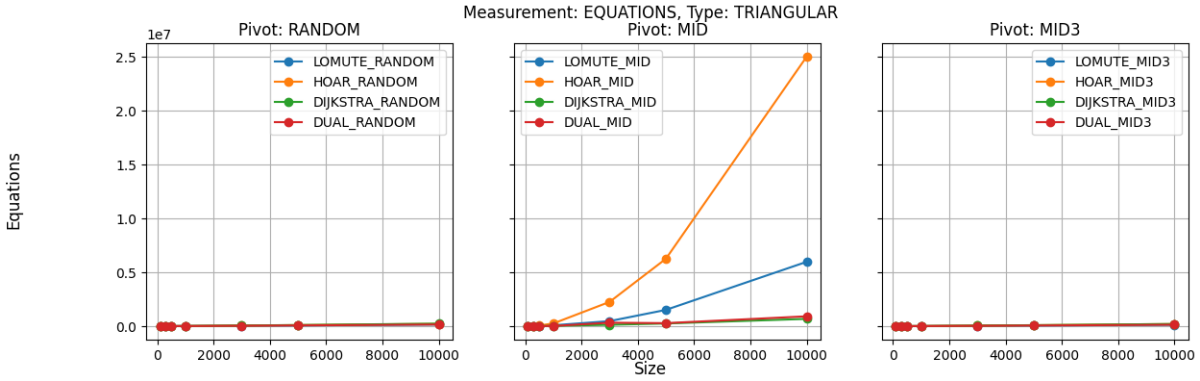
4.11 Результати для "трикутних" масивів:

Час виконання:

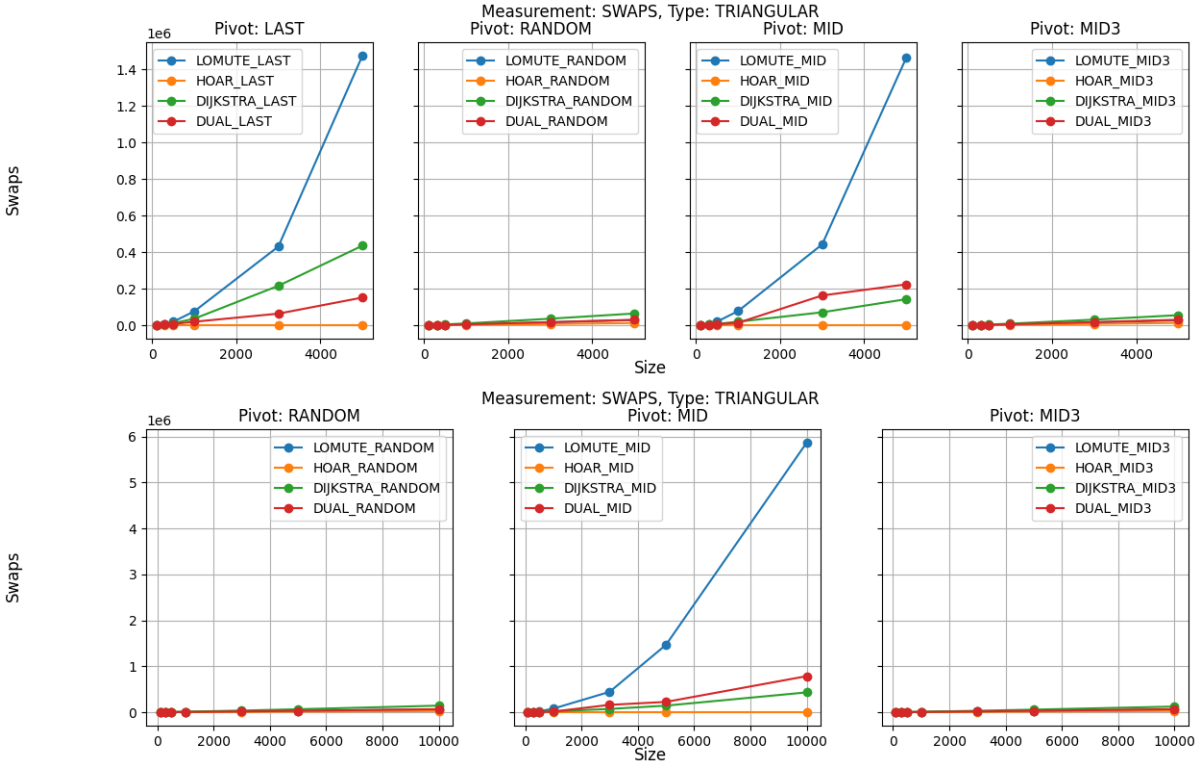


Кількість проведених порівнянь:

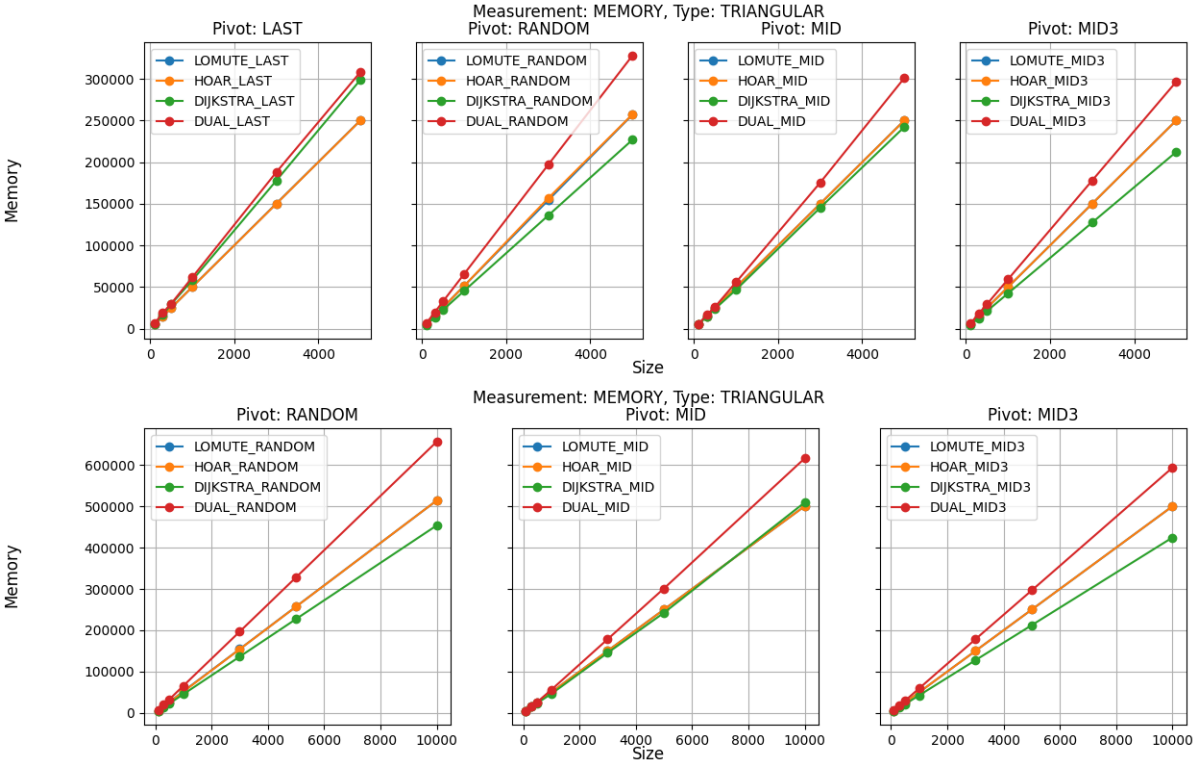




Операцій обмінів:



Використано пам'яті:



4.12 Висновки для "трикутних" масивів:

1. У таких масивів обрання опорного елемента як медіани першого, середнього та останього елемента обирає останній елемент.

2. Обрання опорного елемента як останього приводить до великого часу виконання, великої кількості порівнянь та обмінів.