**Quick Sort**

Randomized: Время увеличивается почти линейно с размером массива. Начиная с 7 мс для 100 элементов, достигает до 664 мс для 5000 элементов. Это указывает на хорошую производительность в среднем случае для быстрой сортировки.

Reverse: Время увеличивается более круто по сравнению со случайными данными. Начинается с 41 мс для 100 элементов и увеличивается до 62973 мс для 5000 элементов. Указывает на то, что быстрая сортировка менее эффективна для данных в обратном порядке.

Almost Sorted: Показывает лучшую производительность, чем данные в обратном порядке, но немного хуже, чем случайные данные. Начинается с 11 мс для 100 элементов, достигая до 12278 мс для 5000 элементов. Быстрая сортировка достаточно хорошо справляется с почти отсортированными данными.

**Quick Heap Sort**

Все варианты показывают похожие графики, но работают разной степенью эффективности.Время увеличивается с размером массива.Увеличение более заметно для данных в обратном порядке и почти отсортированных данных.

Производительность в случайных и почти отсортированных данных лучше, чем в данных в обратном порядке.

В данных в обратном порядке время увеличивается значительно, что указывает на то, что это сложный сценарий для быстрой пирамидальной сортировки.

Эффективность немного варьируется между разными размерами кучи (5, 10, 20, 50), но следует похожей общей тенденции.

**Итог**

Быстрая Сортировка эффективна для случайных и почти отсортированных данных, но менее эффективна для данных в обратном порядке.

Быстрая Пирамидальная Сортировка в различных вариантах показывает последовательное увеличение времени с размером массива, при этом данные в обратном порядке занимают больше всего времени.