For an Array of 7, the Merge Sort went into 2 depths of recursion. And had 18 function calls. (Excluding the initial call to mergeSort).

For an Array of 7, the Quick Sort went into 5 depths of recursion. And had 18 function calls. (Excluding the initial call to quickSort).

This means that while the QuickSort (leftmost pivot) may be more demanding in space as it occupies more in the CallStack. Which means it might be slower.

For the MergeSort in Ascending for an Array of 7 Elements it takes 801.5 nano seconds.

For the QuickSort in Ascending for an Array of 7 Elements it takes 860.3 nano seconds.

For the MergeSort in Ascending for an Array of 100 Elements it takes 3574.2 nano seconds.

For the QuickSort in Ascending for an Array of 100 Elements it takes 10157.3 nano seconds.

This cost is also exacerbated because we use the LeftMostPivot version of the quickSort which isn’t the most efficient version of the quickSort.

Quick Sort degrades to O(n^2) because of the combination of a bad pivot and already sorted data.