1. Merge Sort:
2. Million Random:

Start: 15:06:07.727

End: 15:06:08.5

Time Taken: 0.773 sec

1. 10 Million Random:

Start: 16:21:34.9

End: 16:21:35.991

Time Taken: 1.091 sec

Algorithm Description: Merge Sort is a divide and conquer algorithm. It makes a copy of the input list and divides the copied list in half and uses recursion to continuously divide the list in halves until there is only a single element in the list, which is considered sorted. It then moves back up the levels of divided lists and merges them together sorting the elements level by level until it makes its way up to the first list. The worst, best and average case scenarios are O(nLogn) this is because no matter what merge sort will always divide the list in half until there are only single elements and merge them back up. When it comes to large list merge sort is efficient because the O(nLogn) allows it to take fewer overall steps to sort the list unlike other sorts like bubble or selection. However, because it creates a copy of the array it does use more memory.

1. Quick Sort:
2. Million Random:

Start: 16:40:41.272

End: 16:40:41.415

Time Taken: 0.143 sec

1. 10 Million Random:

Start: 16:42:07.279

End: 16:42:08.775

Time Taken: 1.456 sec

Algorithm Description: Like Merge Sort, Quick Sort is a Divide and Conquer algorithm. It picks an element as pivot and partitions the given array around the picked pivot. But unlike Merge sort it does not require another array for it to work. It works by selecting a pivot index from the array and partitioning the other elements into two sub-arrays, according to whether they are less than or greater than the pivot. The sub-arrays are then sorted recursively by partitioning them into smaller arrays and swapping them based on the comparison with 'pivot' element selected. The worst case occurs when the partition process always picks greatest or smallest element as pivot O(n^2). The best and average case is O(nLogn). Merge sort uses extra space due to creating a copy array, Quick sort requires little space.