# **Quiz 11:**

The theoretical time complexity of the implemented algorithm:

n is the size of the given array.

Worst-case:O(n^2)

By choosing the largest or smallest element as the pivot element, after positioning the element after the first iteration it will get the unsorted sub-list of n-1 elements and again in the next iteration it will get the sub-list of n-2 elements. The quick sort will be in the worst case if the array is not evenly divided into two regions.

Average-case: O(nlogn)

When the median or close to it is used as a pivot element in each iteration and the array is divided into almost two equal regions, the quick sort will be in the average case.

Best-case: O(nlogn)

Quick sort will run in the best case if the partitioning procedure produces two regions of size n/2.

The output of testing the implemented algorithm:

Test Case1:

Before Sort:[95, 6, 30, 99, 33, 86, 43, 1, 96, 97]

After Sort: [1, 6, 30, 33, 43, 86, 95, 96, 97, 99]

Test Case2:

Before Sort:[30, 13, 22, 21, 18, 85, 88, 65, 82, 93, 90, 0, 61, 91, 38, 20, 98, 47, 20, 84]

After Sort: [0, 13, 18, 20, 20, 21, 22, 30, 38, 47, 61, 65, 82, 84, 85, 88, 90, 91, 93, 98]

Test Case3:

Before Sort:[17, 43, 89, 85, 16, 80, 42, 41, 46, 67, 4, 83, 59, 85, 90, 23, 16, 95, 76, 76]

After Sort: [4, 16, 16, 17, 23, 41, 42, 43, 46, 59, 67, 76, 76, 80, 83, 85, 85, 89, 90, 95]