# PROJECT 3

**QUICK SORT and ALGORITHM ANALYSIS**

**(15+30+15)**

Using Quick Sort algorithms implement a Program in Java or Python to sort an array of real numbers of size N.

Do all part A, B, and C.

A. T(n)= Theta (n\*n)

1. Input at least 2 or more sets of data with at least n real numbers in each. For example, n= 10, and 25.
2. Input array data must be good for Worst Case Quick Sort [T(n)= Theta (n\*n)].
3. Display the array after each partition function for the Worst Case Complexity of the quick sort algorithm including the display of the beginning array data input and the ending array data output for each array of n elements.
4. Display a table of n, Actual count for T(n), and T(n)= Theta (n \* n).

B. T(n)= Theta (n lg n).

1. Input at least 2 or more sets of data with at least n real numbers in each. For example, n= 10, and 25.
2. Input array data must be good for general Case Quick Sort [T(n)= Theta (n lgn)].
3. Display the array after each partition function for the general Case Complexity of the quick sort algorithm including the display of the beginning array data input and the ending array data output for each array of n elements.
4. Display a table of n, Actual count for T(n) , and T(n)= Theta (n lg n).

C. C A R Hoare

1. Use the original version (1962) of Quick Sort by C A R Hoare which is listed in page 185, problem 7.1 of Cormen’s book 3rd edition and do part B again.

(4th edition page 199)

1. Professional output with necessary information for readers are required.