**INSERTION SORT, MERGE SORT and ALGORITHM ANALYSIS**

1. Using Insertion Sort algorithm Implement a Program in Python language you desire to sort an array of Real Numbers of size N.
2. Input at least 10 or more sets of randomized unsorted data with at least N elements in each set. For example, N= 12, 18, 24, 30, 36, 42, 48, 54, 60, 72.
3. Display the Unsorted data input and sorted data output for each array of N elements. Plot a graph to compare the worst case (average case) algorithm and the actual count putting counters in strategic points of your programs by assuming the cost ci = 1 for all statements.
4. Display a table of N, Actual count, and the worst case T(N).
5. Plot a graph to compare the Worst Case Complexity of the algorithm and actual count putting counters in strategic points of your programs. Input data must be good for Worst Case Insertion Sort or average case insertion sort.

Notes: The axis of the graph should be the theoretic total cost T(N) or the actual count vs. the N value.

1. Using Merge sort algorithm Implement a Program in the language you select to sort an array of Real numbers of size N.