# Task: Sorting Algorithms Implementation and Performance Analysis.

the task consists of two stages:

Stage 1:

1. Implement the following sorting algorithms:

a) Counting Sort

b) Quick Sort

c) Bubble Sort

d) Merge Sort

e) Radix Sort

Input: Unsorted list of elements.

Output: Sorted list in ascending order.

2. Utilize the provided test function, **test\_sorting\_algorithm(sort\_func)**, to verify the correctness of your sorting functions. This function will run random test cases and compare the sorted output with the expected result.

3. Use the provided plotting function, **plot\_sorting\_performance(algorithms)**, to analyze the performance of the sorting algorithms. The input to this function should be a **dictionary** containing the names of the sorting algorithms and their corresponding function references.

Stage 2:

Extend each of the sorting algorithms from Stage 1 to include a counter that tracks the number of operations performed during the sorting process. An operation can be defined as a comparison or a swap between elements.

Additionally, modify the plot\_sorting\_performance(algorithms) function to incorporate the operation counters for each sorting algorithm. The modified function should plot the performance of the sorting algorithms in terms of the number of operations as a function of the length of the input array.