**1.Understand Sorting Algorithms:**

***Q1.Explain different sorting algorithms (Bubble Sort, Insertion Sort, Quick Sort, Merge Sort).***

**Bubble Sort:**

This is a very simple comparison-based sorting algorithm. In this process, the repetition through the list, comparing adjacent elements, and permuting them if they are in the wrong order takes place until the list is sorted.

**Time Complexity:**

Best Case: O(n)

Average Case: O(n^2)

Worst Case: O(n^2)

**Insertion Sort:**

Description: Generates the final sorted array one element at a time. It selects the next element and fits it into the proper position by comparing it with the sorted elements.

**Time Complexity:**

Best Case: O(n)

Average Case: O(n^2)

Worst Case: O(n^2)

**Quick Sort:**

Description: It is a divide and conquer algorithm that picks an element as a 'pivot', partitions the given array around the picked pivot, there are other two arrays, one with elements smaller than the picked pivot and another with elements greater than the picked pivot. Finally, it recursively sorts the sub-arrays.

**Time Complexity:**

Best Case: O(n log n)

Average Case: O(n log n)

Worst Case: O(n^2)

**Merge Sort:**

Description: It is also a divide-and-conquer algorithm that divides the array into two halves; it recursively sorts them and then merges the sorted halves.

**Time Complexity:**

Best Case: O(n log n)

Average Case: O(n log n)

Worst Case: O(n log n)

**2.Analysis**

***Q.Compare the performance (time complexity) of Bubble Sort and Quick Sort.***

Bubble Sort:

Best Case: O(n) (already sorted)

Average Case: O(n^2)

Worst Case: O(n^2) (reverse sorted)

Quick Sort:

Best Case: O(n log n)

Average Case: O(n log n)

Worst Case: O(n^2) (if choice of the pivot is bad, say always picking up the largest or smallest element)

Comparison:

The algorithm is easy to implement but inefficient on large sets due to its average and worst-case complexity O(n^2).

Quick Sort: A little more complex, usually much faster (on average time complexity O(n log n)). The worst case time complexity is O(n^2), but this can be avoided using good strategies of choosing pivots.

***Q.Discuss why Quick Sort is generally preferred over Bubble Sort.***

Quick sort is generally preferred over bubble sort when one has to deal with large volumes of data since it performs far better in the average case. Bubble sort can nevertheless be useful in at least two scenarios: educational and small datasets where implementation simplicity trumps performance.