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

**A.** Bubble Sort*:*

Bubble Sort is a simple comparison-based sorting algorithm. It repeatedly steps through the list, compares adjacent elements, and swaps them if they are in the wrong order. This process is repeated until the list is sorted.

Insertion Sort:

Insertion Sort is another simple comparison-based sorting algorithm. It builds the final sorted array one item at a time. It is much less efficient on large lists than more advanced algorithms.

Quick Sort:

Quick Sort is a highly efficient comparison-based sorting algorithm. It works by selecting a 'pivot' element from the array and partitioning the other elements into two sub-arrays according to whether they are less than or greater than the pivot.

Merge Sort:

Merge Sort is a stable, comparison-based, divide-and-conquer sorting algorithm. It works by dividing the unsorted list into n sublists, each containing one element, and then repeatedly merging sublists to produce new sorted sublists.

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

**A.** Bubble Sort:

* Time Complexity:
  + Best Case: O(n)
  + Average/Worst Case: O(n^2)

Quick Sort:

* Time Complexity:
  + Best/Average Case: O(n log n)
  + Worst Case: O(n^2) (rare, can be mitigated with good pivot selection)

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

**A.** Quick Sort is preferred because of the following reasons:

* Efficiency: Quick Sort’s average time complexity of O(n log n) is significantly better than Bubble Sort’s O(n²). This makes Quick Sort more suitable for handling large datasets.
* Divide and Conquer: Quick Sort effectively partitions the data and sorts the partitions, leading to faster performance on average.
* In-Place Sorting: Quick Sort requires less additional memory compared to Merge Sort, which can be advantageous in memory-limited environments.