**Understand Sorting Algorithms:**

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

Answer:

**Bubble Sort**

Bubble sort is a simple sorting algorithm that repeatedly steps through the list, compares adjacent elements, and swaps them if they are in the wrong order. The pass through the list is repeated until the list is sorted.

**Time Complexity:** O(n^2)

**Insertion Sort**

Insertion sort is a simple sorting algorithm that works by dividing the input into a sorted and an unsorted region. Each subsequent element from the unsorted region is inserted into the sorted region in its correct position.

**Time Complexity:** O(n^2)

**Quick Sort**

Quick sort is a divide-and-conquer algorithm that selects a pivot element, partitions the array around it, and recursively sorts the sub-arrays.

**Time Complexity:** O(n log n) on average, O(n^2) in the worst case

**Merge Sort**

Merge sort is a divide-and-conquer algorithm that divides the array into two halves, recursively sorts them, and merges them back together in sorted order.

**Time Complexity:** O(n log n)

**Analysis:**

**Q1. Compare the performance (time complexity) of Bubble Sort and Quick Sort.**

Answer:

**Performance Comparison**

| **Algorithm** | **Time Complexity** |
| --- | --- |
| Bubble Sort | O(n^2) |
| Quick Sort | O(n log n) on average, O(n^2) in the worst case |

**Q2. Discuss why Quick Sort is generally preferred over Bubble Sort.**

Answer: Quick Sort is generally preferred over Bubble Sort because of its average-case time complexity of O(n log n), which is much faster than Bubble Sort's O(n^2) time complexity. However, in the worst case, Quick Sort's time complexity can degrade to O(n^2), which is similar to Bubble Sort's performance.

In conclusion, Quick Sort is a more efficient sorting algorithm than Bubble Sort, especially for large datasets. However, it's essential to consider the specific requirements and constraints of the problem before choosing a sorting algorithm.