1. Understand Sorting Algorithms:

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

### Ans:Bubble Sort

A simple comparison-based algorithm that repeatedly steps through the list, compares adjacent items, and swaps them if they are in the wrong order. This process is repeated until no more swaps are needed. Suitable for small datasets or educational purposes due to its simplicity.

Time Complexity: O(n^2)

Insertion Sort

A comparison-based algorithm that builds the sorted list one item at a time by removing each item from the input data and finding the location it belongs in the sorted list. It inserts the item into the correct position. Efficient for small datasets or partially sorted data.

Time Complexity: O(n^2)

Quick Sort

A comparison-based, divide-and-conquer algorithm that picks an element as a pivot and partitions the array around the pivot. The sub-arrays are then sorted recursively. Generally fast and efficient for large datasets; performs well in practice with a good pivot selection strategy.

Time Complexity: O(n logn)

Merge Sort

A comparison-based, divide-and-conquer algorithm that divides the array into halves, recursively sorts each half, and then merges the sorted halves to produce the sorted array. Suitable for large datasets and stable sorts; requires additional memory for merging.

Time Complexity: O(n logn)

4. Analysis:

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

Ans: Bubble Sort

Best case – O(n), Average case – O(n^2), Worst case - O(n^2)

Quick Sort

Best case – O(n log n), Average case – O(n log n) , Worst case - O(n^2)

Discuss why Quick Sort is generally preferred over Bubble Sort.

Ans: **Quick Sort** is generally preferred over **Bubble Sort** due to its efficient average-case time complexity of O(n log n), its ability to handle large datasets more effectively, and its practical applicability in real-world scenarios. **Bubble Sort** is simpler but inefficient for larger datasets, making it less suitable for performance-critical applications.