Solution to Understanding Sorting Algorithms

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

Ans.

1. Bubble Sort: A Simple Comparison-Based Algorithm

Bubble Sort works by repeatedly stepping through the list, comparing adjacent elements, and swapping them if they are in the wrong order. This process is repeated until the list is sorted. Despite its simplicity, Bubble Sort has a worst-case time complexity of O(n^2), making it inefficient for larger datasets.

2. Insertion Sort: Building the Final Sorted List One Element at a Time

Insertion Sort constructs the final sorted list by taking each element and inserting it into the correct position in the already sorted part of the list. It performs well for small datasets and partially sorted lists, but it also has a worst-case time complexity of O(n^2).

3. Quick Sort: Divide and Conquer Approach

Quick Sort selects a pivot element and partitions the list into two sub-lists around the pivot. The two sub-lists are then recursively sorted. On average, Quick Sort has a time complexity of O(n log n), making it efficient for most cases. However, it can degrade to O(n^2) in the worst-case scenario.

4. Merge Sort: Another Divide and Conquer

Merge Sort is a more efficient algorithm based on the divide-and-conquer approach. It divides the list into halves, sorts them separately, and then merges the sorted halves to obtain the final sorted list. Merge Sort has a time complexity of O(n log n), making it more suitable for larger datasets.