Sorting Customer Orders

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

Ans.

Bubble Sort

Bubble Sort is a simple comparison-based algorithm that repeatedly steps through the list, compares adjacent elements, and swaps them if they are in the wrong order. This process is repeated until no more swaps are needed. It has best case time complexity of O(n) when the list is already sorted and worst-case time complexity of O(n²).

Insertion Sort

Insertion Sort builds the final sorted array one element at a time. It picks an element from the unsorted portion and places it in its correct position in the sorted portion. It has best case time complexity of O(n) when the list is already sorted and worst-case time complexity of O(n²).

Quick Sort

Quick Sort is a highly efficient divide-and-conquer algorithm. It selects a 'pivot' element and partitions the array into two sub-arrays, elements less than the pivot and elements greater than the pivot and then recursively sorts the sub-arrays. It has best and average case time complexity of O(n log(n)) while worst case of O(n²) if the pivot selection is poor, like smallest or largest in sorted order.

Merge Sort

Merge Sort is another efficient divide-and-conquer algorithm. It divides the list into equal halves, sorts them recursively, and then merges the sorted halves. It has best, average and worst-case time complexity of O(nlog(n)) and a space complexity of O(n).

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

Ans. In the best case, Bubble Sort has a time complexity of O(n) when the array is already sorted, while Quick Sort performs at O(n log n) even in its best case. For the average case, Bubble Sort operates at O(n²) due to repeated comparisons and swaps, whereas Quick Sort is far more efficient with an average time complexity of O(nlog(n)), making it suitable for larger datasets. In the worst-case scenario, both algorithms have a time complexity of O(n²); however, this is uncommon for Quick Sort if a good pivot selection strategy is used

1. Discuss why Quick Sort is generally preferred over Bubble Sort?

Ans. Quick Sort is generally preferred over Bubble Sort because it is much faster, with an average time complexity of O(n log n) compared to Bubble Sort’s O(n²). It handles large, unsorted data efficiently using a divide-and-conquer approach and requires minimal extra memory. While both have a worst-case of O(n²), Quick Sort avoids it in practice with good pivot selection, making it far more suitable for real-world applications, whereas Bubble Sort is mainly used for teaching basic concepts.