Sorting Customer Orders

Understand Sorting Algorithms:

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

Ans:

Bubble Sort: Simple, compares adjacent elements, $O(n^2)$ average/worst-case, O(1) space. Inefficient for large datasets.

Insertion Sort: Builds sorted array incrementally, $O(n^2)$ average/worst-case, O(1) space. Efficient for small or nearly sorted data.

Quick Sort: Divide-and-conquer, O(n log n) average-case, $O(n^2)$ worst-case, O(log n) space. Fast for large datasets.

Merge Sort: Divide-and-conquer, O(n log n) for all cases, O(n) space. Consistent performance but requires extra space.

Analysis:

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

Ans:

Quick Sort generally outperforms Bubble Sort due to its O(n log n) average-case time complexity, compared to Bubble Sort's $O(n^2)$. While Quick Sort is

faster and more efficient for large datasets, Bubble Sort's O(n) best-case is only ideal for already sorted arrays.

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

Ans:

Quick Sort is preferred over Bubble Sort because it offers significantly better performance with an average-case time complexity of O(n log n), compared to Bubble Sort's $O(n^2)$. Quick Sort efficiently handles large datasets and generally performs faster, whereas Bubble Sort is less efficient and suitable only for small or nearly sorted arrays.