**Exercise 3: Sorting Customer Orders**

**Understand Sorting Algorithms:**

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

Bubble Sort: A sorting algorithm that compares and switches nearby elements if they are out of order for each pair of elements. Until the list is sorted, these steps are repeated. When the list is already sorted, the best case is O(n) and the worst and average cases are O(n^2).

Insertion Sort : Insertion Sort builds the sorted array one element at a time by selecting the subsequent element and inserting it into the appropriate place among the components that have already been sorted. When the list is already sorted, the best case is O(n) and the worst and average cases are O(n^2).

Quick Sort is a divide-and-conquer method that divides an array into two sub-arrays: elements less than the pivot and elements bigger than the pivot. It starts by choosing a "pivot" element. After that, the sub-arrays are sorted recursively. O(n log n) in the best scenario, O(n^2) in the worst (unusual, but manageable with careful pivot selection), and O(n log n) on average.

Merge Sort: This divide-and-conquer algorithm splits the array in half, sorts the halves recursively, and then combines the sorted halves to create the sorted array. O(n log n) is the time complexity in all scenarios (worst, average, and best).

**Analysis:**

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

Answer –

Bubble Sorting - O(n) in the best scenario, O(n^2) in the average and worst cases. Bubble Sort frequently compares and switches nearby components, which typically results in quadratic time complexity.

Quick Sort -  
O(n log n) in the best scenario, O(n^2) in the worst, and O(n log n) on average. Quick Sort employs a divide-and-conquer strategy, which divides the array into sub-arrays and sorts them recursively to typically yield logarithmic performance.

**Question - Discuss why Quick Sort is generally preferred over Bubble Sort.**

Answer – Quick Sort has an average-case time complexity of O(n log n), it is typically substantially faster than Bubble Sort. Bubble Sort is not viable for large datasets due to its quadratic time complexity. Quick Sort uses a more effective divide-and-conquer technique, it scales better with larger arrays. Quick Sort is a better option for sorting operations since, in most cases, its performance is closer to its average-case complexity, especially when coupled with effective pivot selection algorithms.