**Sorting Customer Orders**

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

* Bubble Sort:

1. Description: Repeatedly swaps adjacent elements if they are in the wrong order.
2. Time Complexity: O(n^2) in the worst and average cases; O(n) if the list is already sorted.

* Insertion Sort:

1. Description: Builds the sorted list one item at a time, inserting each item into its correct position.
2. Time Complexity: O(n^2) in the worst and average cases; O(n) in the best case (already sorted).

* Quick Sort:

1. Description: Divides the list into smaller sublists based on a pivot element and recursively sorts them.
2. Time Complexity: O(n^2) in the worst case; O(n log n) on average.

* Merge Sort:

1. Description: Divides the list into halves, recursively sorts them, and merges the sorted halves.
2. Time Complexity: O(n log n) in all cases.
3. Compare the performance (time complexity) of Bubble Sort and Quick Sort

* Bubble Sort:

1. Time Complexity: O(n^2)
2. Advantages: Simple to implement.
3. Disadvantages: Inefficient for large datasets.

* Quick Sort:

1. Time Complexity: O(n log n) on average.
2. Advantages: Efficient for large datasets.
3. Disadvantages: Can degrade to O(n^2) in the worst case, but this is rare with good pivot choices.
4. Why Quick Sort is generally preferred over Bubble Sort  
     
   Quick Sort is generally preferred over Bubble Sort due to its superior average-case time complexity (O(n log n) vs. O(n^2)), making it more efficient for large datasets.