**Exercise 3: Sorting Customer Orders**

**1. Understand Sorting Algorithms:**

* **Bubble Sort**: Repeatedly steps through the list, compares adjacent elements, and swaps them if they are in the wrong order. Time Complexity: O(n^2).
* **Insertion Sort**: Builds the sorted array one item at a time, inserting each item into its correct position. Time Complexity: O(n^2).
* **Quick Sort**: Divides the array into sub-arrays based on a pivot, then recursively sorts the sub-arrays. Time Complexity: O(n log n) on average.
* **Merge Sort**: Divides the array into halves, recursively sorts them, and then merges the sorted halves. Time Complexity: O(n log n).

**4. Analysis:**

* **Bubble Sort vs. Quick Sort**:
  + **Bubble Sort**: O(n^2) time complexity. Simple but inefficient for large datasets.
  + **Quick Sort**: O(n log n) time complexity on average. More efficient and preferred for large datasets due to its faster average performance.

Quick Sort is generally preferred over Bubble Sort because of its better average time complexity, making it more suitable for large datasets.