**Sorting Customer Orders**

1. **Sorting Algorithms**:
   * **Bubble Sort**: A simple sorting algorithm that repeatedly steps through the list, compares adjacent elements, and swaps them if they are in the wrong order. Time complexity is O(n²) in both average and worst cases.
   * **Quick Sort**: A more efficient sorting algorithm that uses divide-and-conquer. It partitions the array into smaller sub-arrays and sorts them recursively. Time complexity is O(n log n) on average and O(n²) in the worst case.
2. **Implementation**:
   * **Bubble Sort**: Implemented in the bubbleSort method, which sorts the orders in ascending order based on the total price.
   * **Quick Sort**: Implemented in the quickSort method, with a partition helper method to perform the partitioning of the array.
3. **Analysis**:
   * **Bubble Sort**: Inefficient for large datasets due to its O(n²) time complexity. It is simple but not suitable for performance-critical applications.
   * **Quick Sort**: Generally preferred over Bubble Sort due to its average O(n log n) time complexity. It is more efficient for large datasets.

Quick Sort is generally preferred over Bubble Sort due to its better average time complexity, making it more suitable for handling larger datasets efficiently.