**Exercise 3: Sorting Customer Orders**

**Scenario:**

You are tasked with sorting customer orders by their total price on an e-commerce platform. This helps in prioritizing high-value orders.

**Steps:**

1. **Understand Sorting Algorithms:**
   * Explain different sorting algorithms (Bubble Sort, Insertion Sort, Quick Sort, Merge Sort).

**Bubble Sort:** Simple, compares each pair of adjacent elements and swaps them if they are in the wrong order. Time complexity: O(n²).

**Insertion Sort:** Builds a sorted array one item at a time, inserting each item into its correct position. Time complexity: O(n²).

**Quick Sort:** Divides the array into smaller sub-arrays based on a pivot and recursively sorts them. Time complexity: O(n log n) on average, O(n²) in the worst case.

**Merge Sort:** Divides the array into halves, sorts each half, and then merges the sorted halves. Time complexity: O(n log n).

1. **Analysis:**
   * **Compare the performance (time complexity) of Bubble Sort and Quick Sort**

**Performance Comparison:**

* **Bubble Sort:** O(n²), not efficient for large datasets due to its quadratic time complexity.
* **Quick Sort:** O(n log n) on average, more efficient than Bubble Sort for large datasets. However, it has a worst-case time complexity of O(n²), which can be mitigated by using random pivots or other optimizations.
  + Discuss why Quick Sort is generally preferred over Bubble Sort.

Quick Sort is generally preferred due to its average-case time complexity of O(n log n) and its in-place nature (it does not require additional memory like Merge Sort). It is faster and more efficient for large datasets compared to Bubble Sort.