**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:**

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

**Answer:**

**Bubble Sort** - is a simple sorting algorithm that works by repeatedly iterating through the data, comparing adjacent elements and swapping them if they are in the wrong order. This process continues until no more swaps are needed, indicating that the data is sorted. Bubble Sort has a time complexity of O(n^2), making it inefficient for large datasets.

**Insertion Sort** - is another simple sorting algorithm that works by iterating through the data one element at a time, inserting each element into its proper position in the sorted portion of the data. Insertion Sort has a time complexity of O(n^2), making it inefficient for large datasets. However, it is more efficient than Bubble Sort for small datasets.

**Quick Sort** - is a divide-and-conquer sorting algorithm that works by selecting a pivot element, partitioning the data around the pivot, and recursively sorting the subarrays. The pivot element is chosen such that all elements less than the pivot are on the left and all elements greater than the pivot are on the right. Quick Sort has an average time complexity of O(n log n), making it a efficient choice for large datasets. However, it can have a worst-case time complexity of O(n^2) if the pivot is chosen poorly.

**Merge Sort** - is another divide-and-conquer sorting algorithm that works by dividing the data into smaller chunks, sorting each chunk, and merging the sorted chunks back together. Merge Sort has a time complexity of O(n log n), making it a efficient choice for large datasets. It is also a stable sorting algorithm, meaning that the order of equal elements is preserved.

**4. Analysis:**

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

**Answer:**

Bubble Sort has a time complexity of O(n^2), which makes it inefficient for large datasets. Quick Sort, on the other hand, has an average time complexity of O(n log n), making it a more efficient choice for sorting large datasets.

**(ii) Discuss why Quick Sort is generally preferred over Bubble Sort.**

**Answer:**

Bubble Sort has a time complexity of O(n^2), which makes it inefficient for large datasets. Quick Sort, on the other hand, has an average time complexity of O(n log n), making it a more efficient choice for sorting large datasets. However, Quick Sort can have a worst-case time complexity of O(n^2) if the pivot is chosen poorly.