**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).**

1. Bubble Sort: A sorting algorithm that repeatedly steps through the list to be sorted, compares each pair of adjacent terms, and swaps them if they are in wrong order. Two adjacent elements of a list are first checked and then swapped.
2. Insertion Sort: A simple sorting algorithm that works by iteratively inserting each element of an unsorted list into its correct position in a sorted portion of the list. It is a stable, in-place sorting algorithm, meaning that elements with equal values maintain their relative order in the sorted output and doesn’t require any additional memory space beyond the original array.
3. Quick Sort: A sorting algorithm based on the Divide and Conquer algorithm that picks an element as a pivot and partitions the given array around the picked pivot by placing the pivot in its correct position in the sorted array.
4. Merge Sort: Merge sort is a sorting algorithm that follows the divide-and-conquer approach. It works by recursively dividing the input array into smaller subarrays and sorting those subarrays then merging them back together to obtain the sorted array.
5. **Setup:**
   * **Create a class Order with attributes like orderId, customerName, and totalPrice.**
6. **Implementation:**
   * **Implement Bubble Sort to sort orders by totalPrice.**
   * **Implement Quick Sort to sort orders by totalPrice.**
7. **Analysis:**

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

1. Bubble Sort :-

* Best Case: O(n)
* Average Case: O(n^2)
* Worst Case: O(n^2)

1. Quick Sort :-

* Best Case: O(n log n)
* Average Case: O(n log n)
* Worst Case: O(n^2)
* **Discuss why Quick Sort is generally preferred over Bubble Sort.**

1. Quick Sort has average time complexity of O(nlogn) which is significantly better than average time complexity of Bubble Sort which has O(n^2).
2. As Quick Sort is a divide-and-conquer algorithm it is efficient on large records than Bubble Sort.