**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 Simple Comparison-Based Algorithm**

Bubble Sort works by repeatedly stepping through the list, comparing adjacent elements, and swapping them if they are in the wrong order. This process is repeated until the list is sorted. Despite its simplicity, Bubble Sort has a worst-case time complexity of **O(n^2)**, making it inefficient for larger datasets.

**2. Insertion Sort: Building the Final Sorted List One Element at a Time**

Insertion Sort constructs the final sorted list by taking each element and inserting it into the correct position in the already sorted part of the list. It performs well for small datasets and partially sorted lists, but it also has a worst-case time complexity of **O(n^2)**.

**3. Quick Sort: Another Divide and Conquer Approach**

Quick Sort selects a pivot element and partitions the list into two sub-lists around the pivot. The two sub-lists are then recursively sorted. On average, Quick Sort has a time complexity of O(n log n), making it efficient for most cases. However, it can degrade to **O(n^2)** in the worst-case scenario.

**4. Merge Sort: Divide and Conquer**

Merge Sort is a more efficient algorithm based on the divide-and-conquer approach. It divides the list into halves, sorts them separately, and then merges the sorted halves to obtain the final sorted list. Merge Sort has a time complexity of **O(n log n),** making it more suitable for larger datasets.

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

|  |  |  |  |
| --- | --- | --- | --- |
| **Algorithm** | **Best-case** | **Average-case** | **Worst-case** |
| Bubble Sort | O(n) | O(n^2) | O(n^2) |
| Quick Sort | O(n log n) | O(n log n) | O(n^2) |

* + Discuss why Quick Sort is generally preferred over Bubble Sort.

Quick Sort is generally prepared over the Bubble sort because of the three best functionality of the quick sort, it is a faster performance and stability also the scalability. Quicksort has a better time complexity than bubble sort, making more efficient for large data set model and it also a stable sorting algorithm which means that the order of equal elements is preserved. To set a large data set and its schedule for sorting messy amounts of data on the other side power series is the simple limit but has a poor time complexity making it less suitable for large data set.