## Sorting Customer Orders

1. Understand Sorting Algorithms:

* **Bubble Sort** repeatedly steps through the list, compares adjacent elements, and swaps them if they are in the wrong order. It continues this process until the list is sorted. It is simple but inefficient for large datasets.
* **Insertion Sort** builds the sorted array one element at a time by repeatedly taking the next element and inserting it into the correct position among the previously sorted elements. It works well for small or nearly sorted data.
* **Quick Sort** is a divide-and-conquer algorithm that selects a ‘pivot’ element, partitions the array around the pivot so that smaller elements come before it and larger elements after it, then recursively sorts the partitions. It is generally very efficient.
* **Merge Sort** also uses divide-and-conquer by splitting the array into halves, recursively sorting each half, and then merging the sorted halves. It guarantees good performance even for large datasets.

1. Time Complexity Analysis

* **Bubble Sort** has a worst-case and average-case time complexity of **O(n²)**, making it inefficient for large datasets.
* **Quick Sort** has an average-case time complexity of **O(n log n)**, which is significantly faster for large datasets. However, its worst case is O(n²), but this is rare with good pivot selection.

1. **Which is Preferred**

* Quick Sort is generally preferred over Bubble Sort because it sorts large datasets much faster and uses less time on average, making it more practical for real-world applications like sorting customer orders on an e-commerce platform.