Ex 3: Sorting Customer Orders

Bubble Sort:

- Repeatedly compares adjacent elements and swaps them if they are in the wrong order.
- Time Complexity : Average: $O(n^2)$

Insertion Sort:

- Builds the sorted array one item at a time by comparing each new element with the already sorted ones.
- **Time Complexity** : Average: $O(n^2)$

Quick Sort:

- Picks a pivot and partitions the array into two halves: less than and greater than pivot, recursively sorted.
- **Time Complexity** : Average: $O(n \log n)$

Merge Sort:

- Divides the array into halves, sorts them recursively, and then merges the sorted halves.
- **Time Complexity** : $O(n \log n)$ in all cases

Time Complexity:

Bubble Sort:

- Time Complexity: $O(n^2)$
- Suitability : Very inefficient, educational only

Quick Sort:

- Time Complexity : $O(n \log n)$
- Suitability : Fast, efficient, widely used

Why Quick Sort is generally preferred over Bubble Sort?

- Much faster on average ($O(n \log n)$ vs $O(n^2)$).
- More scalable for large datasets.
- Bubble Sort is only good for learning concepts or very small datasets.