**Exercise 3: Sorting Customer Orders**   
  
  
**Understanding Sorting Algorithms:**  
 **Bubble Sort:**  
 Compares adjacent values and swaps them repeatedly .  
 Time Complexity: O(n^2)  
 Ideal for small Datasets  
 **Insertion Sort:**  
 Builds a sorted array at the time of insertion itself.  
 Time Complexity : O(n^2)  
 Ideal for small arrays.|  
 **Quick Sort:**  
 Picks a pivot and partitions into 2 halves and recursively sorts them .  
 Time Complexity: O(n log n)  
 Ideal for large datasets   
 **Merge Sort:**  
 Divide the array , sort the divided array and merges them .  
 Time Complexity: O(n log n)   
 Ideal for large files , linked lists  
  
**Time Complexity Analysis :**  
 Bubble Sort - O(n^2) - Suitable for Small Datasets  
 Quick Sort - O(n log n) - Suitable for Large Datasets  
  
**Why Quick Sort is Preferred:**  
 It is faster compared to other sorts (O (n log n))   
 Efficient memory usage