***Exercise 3: Sorting Customer Orders***

**1. Understand Sorting Algorithms:**

***o Explain different sorting algorithms (Bubble Sort, Insertion Sort, Quick Sort, Merge Sort).***

* ***Bubble Sort:***

It is the simplest sort method which performs sorting by repeatedly moving the largest element to the highest index of the array. It comprises of comparing each element to its adjacent element and replace them accordingly.

* ***Insertion Sort:***

Insertion sortis 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 stablesortingalgorithm, meaning that elements with equal values maintain their relative order in the sorted output.

* ***Quick Sort:***

Quick Sort is a sorting algorithm based on the [Divide and Conquer algorithm](https://www.geeksforgeeks.org/divide-and-conquer-algorithm-introduction/)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.

* ***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.

In simple terms, we can say that the process of merge sort is to divide the array into two halves, sort each half, and then merge the sorted halves back together. This process is repeated until the entire array is sorted.

**4. Analysis:**

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

***Bubble Sort:***

* **Best Case:** O(n) - When the array is already sorted (optimized version).
* **Average Case:** O(n^2) - Requires nested loops to compare and swap elements.
* **Worst Case:** O(n^2) - When the array is sorted in reverse order.

***Quick Sort:***

* **Best Case:** O(nlogn) - When the pivot divides the array into roughly equal halves.
* **Average Case:** O(nlogn) - Generally performs well with a good pivot selection.
* **Worst Case:** O(n^2) - When the pivot is always the smallest or largest element (can be mitigated with randomized or median-of-three pivot strategies).

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

* Bubble Sort algorithm significantly performs slower than Quick Sort for larger array sizes. This is due to the fact that Bubble Sort has a complexity of O(n²), while Quick Sort has a complexity of O(n log n).
* The number of comparisons in Bubble Sort increases rapidly with the array size since each element needs to be compared with other elements.
* Quick Sort algorithm achieves faster sorting by using a pivot element and shows better performance with larger arrays due to its complexity.
* The average case complexity of Quick Sort being O(n log n) indicates that it is a more efficient sorting method for larger arrays.