**Task - 3 : Sorting Customer Orders**

Solution by: Mahika Thakur, CSE 7TH Semester, MCKV Institute of Engineering

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

**Bubble Sort**

Description: Bubble Sort is a simple comparison-based algorithm where each pair of adjacent elements is compared, and the elements are swapped if they are in the wrong order. This process is repeated until the array is sorted.

Time Complexity:

Best Case: O(n) when the array is already sorted.

Average Case: O(n^2)

Worst Case: O(n^2)

Space Complexity: O(1) (in-place sorting)

**Insertion Sort**

Description: Insertion Sort builds the final sorted array one item at a time. It iterates through the array and removes one element, finds the appropriate location within the sorted portion of the array, and inserts it there.

Time Complexity:

Best Case: O(n) when the array is already sorted.

Average Case: O(n^2)

Worst Case: O(n^2)

Space Complexity: O(1) (in-place sorting)

**Quick Sort**

Description: Quick Sort is a divide-and-conquer algorithm. It selects a 'pivot' element and partitions the array into two sub-arrays according to whether the elements are less than or greater than the pivot. It then recursively sorts the sub-arrays.

Time Complexity:

Best Case: O(n log n)

Average Case: O(n log n)

Worst Case: O(n^2) (rare, occurs when the pivot selection is poor)

Space Complexity: O(log n) (for the recursive stack)

**Merge Sort**

Description: Merge Sort is also a divide-and-conquer algorithm. It divides the array into halves, recursively sorts them, and then merges the sorted halves.

Time Complexity:

Best Case: O(n log n)

Average Case: O(n log n)

Worst Case: O(n log n)

Space Complexity: O(n) (for the temporary arrays used in merging)

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

Performance Comparison

Bubble Sort:

Best Case: O(n) - The array is already sorted.

Average Case: O(n^2)

Worst Case: O(n^2)

Quick Sort:

Best Case: O(n log n)

Average Case: O(n log n)

Worst Case: O(n^2) - Rare, occurs with poor pivot selection.

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

Quick Sort is generally preferred over Bubble Sort due to its significantly better average-case time complexity (O(n log n) compared to O(n^2)). Quick Sort is more efficient for large datasets and is faster in practice because of its lower constant factors and better cache performance due to its divide-and-conquer strategy.