Exercise 3: Sorting Customer Orders

a.Understand Sorting Algorithms:

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

Bubble Sort:

I would say that Bubble Sort might be the simplest sorting algorithm. The way this algorithm processes the input is just like a bubble trying to reach out to the surface, within each iteration the algorithm will find the highest value and put it at the end of the data-set or where that value belongs by comparing each pair of elements in the data-set. The pass through the list is repeated until the list is sorted.

Insertion Sort: Insertion sort is 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 stable sorting algorithm, meaning that elements with equal values maintain their relative order in the sorted output.

Quick Sort: It is a faster and highly efficient sorting algorithm based on the Divide and Conquer algorithm 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 one of the most efficient sorting algorithms. It is based on the divide-and-conquer strategy. Merge sort continuously cuts down a list into multiple sublists until each has only one item, then merges those sublists into a sorted list.

b. Analysis:

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

1. Bubble Sort Algorithm:

- Bubble Sort compares neighboring elements and swaps their positions to sort the array.

- It uses two loops to compare all elements and performs necessary swaps.

- The number of comparisons is n * (n — 1) / 2 since each element needs to be compared with other elements.

- Its complexity is $O(n^2)$.

2. Quick Sort Algorithm:

- Quick Sort performs sorting by selecting a pivot element, partitioning the array, and sorting the subarrays.

- It operates by finding the correct position for the pivot element and moving elements to their correct positions.

- Subarrays are recursively sorted.

- The number of comparisons is n * log(n) in the best and average cases, and n^2 in the worst case.

- Its complexity is O(n log n) in the best and average cases, and $O(n^2)$ in the worst case.

2. Discuss why Quick Sort is generally preferred over Bubble Sort.

Quick sort algorithm is preferred over bubble Sort algorithm because bubble sort performs slower than Quick Sort for larger array sizes. This is due to the fact that Bubble Sort has a complexity of $O(n^2)$, while Quick Sort has a complexity of O(n log n).