Exercise 3: Sorting Customer Orders

1. Understand Sorting Algorithms:

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

= 1. Bubble Sort: Repeatedly iterate through the array, comparing adjacent pairs of elements and swapping them if they are in the wrong order. Repeat until the array is fully sorted.

2. Insertion Sort: Build up a sorted subarray from left to right by inserting each new element into its correct position in the subarray. Repeat until the array is fully sorted.

3. Quick Sort: Select a 'pivot' element from the array and partition the other elements into two sub-arrays, according to whether they are less than or greater than the pivot. Recursively apply the same process to the sub-arrays.

4. Merge Sort: Divide the array into two halves, sort each half, and then merge the sorted halves to produce the sorted array.

4. Analysis:

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

=Time Complexity Analysis:

Bubble Sort:

Best-case: O(n)

Average-case: O(n^2)

Worst-case: O(n^2)

Quick Sort:

Best-case: O(nlogn)

Average-case: O(nlogn)

Worst-case: O(n^2)

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

= Suitability: Quick Sort is generally preferred over Bubble Sort because it has a better average-case time complexity of O(nlogn) compared to Bubble Sort's O(n^2).