**Exercise 3: Sorting Customer Orders Theory**

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

**Ans-> Bubble Sort**

* **Description**: Compares adjacent elements and swaps them if they are in the wrong order. This process repeats until the list is sorted.
* **Time Complexity**: O(n^2) in the worst and average cases, O(n) in the best case when the array is already sorted.

**Insertion Sort**

* **Description**: Builds the sorted array one element at a time by repeatedly picking the next element and inserting it into its correct position.
* **Time Complexity**: O(n^2) in the worst and average cases, O(n) in the best case when the array is already sorted.

**Quick Sort**

* **Description**: Divides the array into two smaller sub-arrays based on a pivot element, recursively sorts the sub-arrays, and combines them.
* **Time Complexity**: O(n log n) on average, O(n^2) in the worst case if the pivot elements are poorly chosen.

**Merge Sort**

* **Description**: Divides the array into two halves, recursively sorts them, and then merges the sorted halves.
* **Time Complexity**: O(n log n) in all cases.

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

**Ans-> Time Complexity**

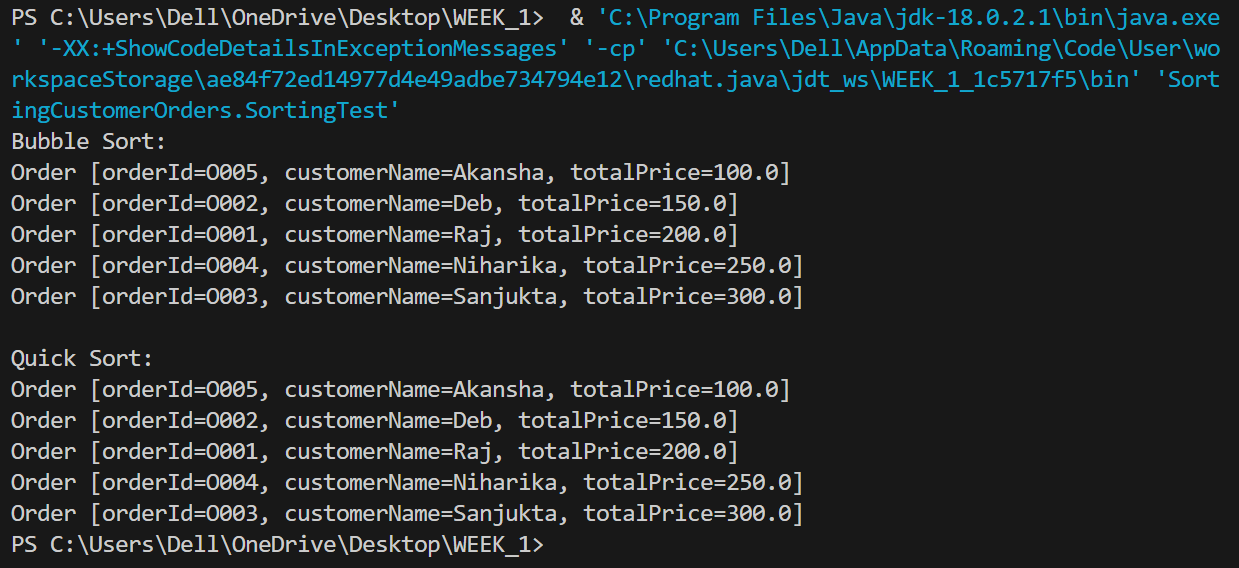
* **Bubble Sort**: O(n^2) in the worst and average cases.
* **Quick Sort**: O(n log n) on average, O(n^2) in the worst case with poor pivot selection.

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

**Ans-> Performance**: Quick Sort generally outperforms Bubble Sort, especially for large datasets, due to its average-case time complexity of O(n log n).

**Efficiency**: Quick Sort is more efficient with memory usage compared to other algorithms like Merge Sort, which requires additional space for merging.

**OUTPUT OF IMPLEMENTATION OF BUBBLE SORT AND QUICK SORT ON TOTAL PRICES-**

****