**Sorting Customer Orders**

**1. Understanding Sorting Algorithms**

**1. Bubble Sort:**

Bubble Sort always compares two-elements positions and swaps them if the first compares less than the second one. This process is done over and over again until the array is sorted.

**2. Insertion Sort:**

Insertion Sort constructs the final sorted array on an item-to-item basis of picking the element and inserting it in its position.

**3. Quick Sort:**

Quick Sort sorts dividend the array and a pivot element and then sort the two resulting arrays. The pivot element is positioned in the order that is needed.

**4. Merge Sort:**

It is a divide and conquer method in which the array is divided into two halves, each of them is sorted in turn and then merged.

**4. Comparison of Bubble sort and Quick sort**

**1. Time Complexity Comparison:**

• Bubble Sort: runs in O(n^2) in the worst case and on average. This implies that the above operation consumes more and more time as the number of elements increases and thus is not suitable for large datasets.

• Quick Sort: is faster than Bubble sort, with an average time-complexity of O (n log n) and is thus better for larger cases of n. But, in the worst case, the time complexity is of order O(n^2) if the pivot is badly chosen though there are mechanisms of achieving randomly worst cases for the selection of the pivot.

**2. Why Quick Sort is Preferred:Why Quick Sort is Preferred:**

Among the methods Quick Sort is better for big datasets as it runs in O(n log n) while Bubble Sort is in O(n^2). It sits within the original memory and requires little extra storage area; and, in fact, one often observes better performances in real use, because there are fewer comparisons and swaps.