**Exercise 3: Sorting Customer Orders**

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

Bubble Sort

Bubble Sort is a simple sorting algorithm that repeatedly steps through the list, compares adjacent elements, and swaps them if they are in the wrong order. The process is repeated until the list is sorted.

* Time Complexity:
  + Best-case: O(n)
  + Average-case: O(n^2)
  + Worst-case: O(n^2)

Insertion Sort

Insertion Sort builds the sorted array one item at a time, taking each new element and inserting it into its correct position.

* Time Complexity:
  + Best-case: O(n)
  + Average-case: O(n^2)
  + Worst-case: O(n^2)

Quick Sort

Quick Sort is a divide-and-conquer algorithm that works by selecting a 'pivot' element from the array and partitioning the other elements into two sub-arrays, according to whether they are less than or greater than the pivot. The sub-arrays are then sorted recursively.

* Time Complexity:
  + Best-case: O(n log n)
  + Average-case: O(n log n)
  + Worst-case: O(n^2)

Merge Sort

Merge Sort is another divide-and-conquer algorithm that divides the array into two halves, sorts them, and then merges them back together.

* Time Complexity:
  + Best-case: O(n log n)
  + Average-case: O(n log n)
  + Worst-case: O(n log n)

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

Time Complexity Comparison

1. Bubble Sort:
   * Best-case: O(n)
   * Average-case: O(n^2)
   * Worst-case: O(n^2)
2. Quick Sort:
   * Best-case: O(n log n)
   * Average-case: O(n log n)
   * Worst-case: O(n^2)

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

* Efficiency: Quick Sort has a better average-case time complexity (O(n log n)) compared to Bubble Sort (O(n^2)), making it more efficient for larger datasets.
* Scalability: Quick Sort performs well on average and is suitable for large arrays, whereas Bubble Sort becomes impractically slow as the number of elements increases.
* Real-world Performance: Quick Sort's divide-and-conquer approach and efficient use of memory often result in better performance in practical scenarios compared to the simpler but less efficient Bubble Sort.