Exercise 3: Sorting Customer Orders

# 1. Understand Sorting Algorithms

## Bubble Sort:

Bubble Sort repeatedly steps through the list, compares adjacent elements, and swaps them if they are in the wrong order. It continues until no swaps are needed. Simple but inefficient.

## Insertion Sort:

Insertion Sort builds the sorted array one element at a time by comparing and inserting elements into their correct position. Efficient for small datasets or nearly sorted data.

## Quick Sort:

Quick Sort is a divide-and-conquer algorithm. It selects a 'pivot' and partitions the array into elements less than and greater than the pivot, then recursively sorts the sub-arrays. Very efficient for large datasets.

## Merge Sort:

Merge Sort is another divide-and-conquer algorithm. It divides the array into halves, recursively sorts them, and then merges the sorted halves. It guarantees O(n log n) time but uses extra space.

# 4. Analysis

## Time Complexity Comparison:

1. Bubble Sort: O(n^2) worst and average case, O(n) best case (already sorted)
2. Quick Sort: O(n log n) average and best case, O(n^2) worst case (rare with good pivot selection)

## Why Quick Sort is Preferred Over Bubble Sort:

* Quick Sort is much faster for large datasets due to its divide-and-conquer approach.
* Bubble Sort is simple but inefficient and only used for educational purposes or very small datasets.