**Understand Sorting Algorithms:**

Sorting is one of the most basic and important operations in computer science. It helps in organizing data, making search and other operations more efficient. There are many sorting algorithms, each with different logic and performance. Below are four commonly used ones:

**Bubble Sort:**  
Bubble Sort is the simplest sorting algorithm. It repeatedly goes through the list, compares adjacent elements, and swaps them if they’re in the wrong order. This process continues until the list is completely sorted. Although easy to understand, it's very slow for large datasets.

**Insertion Sort:**  
Insertion Sort builds the sorted list one item at a time. It picks one element from the unsorted portion and inserts it into the correct position in the sorted part. It’s faster than Bubble Sort for small or nearly sorted data but still inefficient for large datasets.

**Quick Sort:**  
Quick Sort is a divide-and-conquer algorithm. It selects a pivot element and rearranges the array so that elements less than the pivot come before it and elements greater than the pivot come after. Then it recursively sorts the two halves. It’s very efficient and widely used in real-world applications.

**Merge Sort:**  
Merge Sort also follows the divide-and-conquer approach. It splits the array into halves, recursively sorts them, and then merges the sorted halves. It is very stable and consistent in performance but uses more memory than Quick Sort.

**Performance Comparison: Bubble Sort vs. Quick Sort:**  
Bubble Sort has a **time complexity of O(n²)** in both average and worst cases, which means it becomes extremely slow as the size of the input grows. It’s not practical for large datasets. On the other hand, Quick Sort has an **average-case time complexity of O(n log n)**, making it much faster for large data. Even in the worst case, where the pivot selection is poor, Quick Sort has O(n²) complexity—but this can usually be avoided with good pivot selection techniques.

**Why Quick Sort is Preferred:**  
Quick Sort is generally preferred over Bubble Sort because it is much more efficient and handles larger datasets well. While Bubble Sort is simple to understand and implement, it’s too slow for real applications. Quick Sort is faster due to its efficient divide-and-conquer strategy, and it works well even without using extra memory (unlike Merge Sort). That’s why Quick Sort is often used in standard libraries and production-level code.