**Exercise 3:**

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

**Bubble Sort**: Repeatedly compares and swaps adjacent elements if they are in the wrong order.

**Insertion Sort**: Builds the sorted array one element at a time by inserting elements into their correct position.

**Quick Sort**: Divides the array using a pivot and recursively sorts the left and right partitions.

**Merge Sort**: Recursively splits the array into halves and merges them in sorted order.

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

| **Algorithm** | **Best Case** | **Average Case** | **Worst Case** |
| --- | --- | --- | --- |
| **Bubble Sort** | O(n) | O(n²) | O(n²) |
| **Quick Sort** | O(n log n) | O(n log n) | O(n²) |

Discuss why Quick Sort is generally preferred over Bubble Sort.

Quick Sort is faster with an average time complexity of O(n log n), while Bubble Sort takes O(n²).  
It uses a divide-and-conquer strategy, making it efficient for large datasets.  
Quick Sort performs fewer swaps and comparisons than Bubble Sort.  
Due to its speed and scalability, Quick Sort is preferred in real-world applications.

Output:

