1. **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. This process is repeated until the list is sorted.

* **Time Complexity**:
  + **Best Case**: O(n) – When the list is already sorted (optimized version).
  + **Average Case**: O(n^2) – For randomly ordered elements.
  + **Worst Case**: O(n^2) – When the list is sorted in reverse order.

**Insertion Sort**

Insertion Sort builds the final sorted array one item at a time. It picks each item from the input data and places it in the correct position in the sorted list.

* **Time Complexity**:
  + **Best Case**: O(n) – When the list is already sorted.
  + **Average Case**: O(n^2) – For randomly ordered elements.
  + **Worst Case**: O(n^2) – When the list is sorted in reverse order.

**Quick Sort**

Quick Sort is a divide-and-conquer algorithm. It picks an element as a pivot and partitions the array around the pivot. The sub-arrays are then sorted recursively.

* **Time Complexity**:
  + **Best Case**: O(n log n) – When the pivot divides the list into nearly equal parts.
  + **Average Case**: O(n log n) – For randomly ordered elements.
  + **Worst Case**: O(n^2) – When the pivot is the smallest or largest element each time.

**Merge Sort**

Merge Sort is a divide-and-conquer algorithm that divides the list into halves, recursively sorts each half, and then merges the sorted halves to produce the sorted list.

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

**Analysis**

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

1. **Bubble Sort**:
   * **Best Case**: O(n) – When the list is already sorted.
   * **Average Case**: O(n^2) – For randomly ordered elements.
   * **Worst Case**: O(n^2) – When the list is sorted in reverse order.
2. **Quick Sort**:
   * **Best Case**: O(n log n) – When the pivot divides the list into nearly equal parts.
   * **Average Case**: O(n log n) – For randomly ordered elements.
   * **Worst Case**: O(n^2) – When the pivot is the smallest or largest element each time.

**Discuss Why Quick Sort is Generally Preferred Over Bubble Sort**

* **Efficiency**: Quick Sort generally has better average and worst-case time complexities (O(n log n)) compared to Bubble Sort (O(n^2)). This makes Quick Sort more efficient for large datasets.
* **Divide and Conquer**: Quick Sort’s divide-and-conquer approach helps it perform faster by dividing the problem into smaller sub-problems, whereas Bubble Sort repeatedly compares adjacent elements, which can be slower for larger lists.