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

Repeatedly steps through the list, compares adjacent elements, and swaps them if they are in the wrong order. The process continues until no swaps are needed, indicating that the list is sorted.

* 1. Insertion Sort

Builds the sorted array one element at a time by repeatedly picking the next element and inserting it into its correct position among the previously sorted elements.

* 1. Quick Sort

A divide-and-conquer algorithm that selects a 'pivot' element and partitions the array into two sub-arrays according to whether the elements are less than or greater than the pivot. It then recursively sorts the sub-arrays.

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

Bubble Sort

Best Case: O(n)-Occurs when the array is already sorted.

Average Case: O(n^2)-Comparisons and swaps are done in a nested loop for each element.

Worst Case: O(n^2)-Happens when the array is sorted in reverse order

Quick Sort

Best Case: O(n log n)- Occurs when the pivot splits the array into two nearly equal halves.

Average Case: O(n log n)-On average, the pivot splits the array into balanced partitions.

Worst Case: O(n^2)- Happens when the pivot selection is poor, such as always choosing the smallest or largest element as the pivot

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

* It is a divide-and-conquer algorithm that makes it easier to solve problems.
* It is efficient on large data sets.
* It has a low overhead, as it only requires a small amount of memory to function.