**1.**

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

**Ans :** Data structures and algorithms are quite important in the case of huge inventories since they determine the efficiency of data storage, retrieval, and manipulation. Efficient data structures and algorithms ensure that:

* **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. The pass through the list is repeated until the list is sorted.
* **Insertion Sort** builds the final sorted array one item at a time. It is much less efficient on large lists than more advanced algorithms such as quicksort, heapsort, or merge sort.
* **Quick Sort** is an efficient sorting algorithm that uses the divide-and-conquer approach. It works by selecting a 'pivot' element from the array and partitioning the other elements into two sub-arrays according to whether they are less than or greater than the pivot.
* **Merge Sort** is an efficient, stable, comparison-based, divide-and-conquer sorting algorithm. Most implementations produce a stable sort, meaning that the implementation preserves the input order of equal elements in the sorted output.

**4.**

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

**Ans :**

* **Bubble Sort**:

Best case: O(n) (when the list is already sorted)

Average case: O(n^2)

Worst case: O(n^2)

* **Quick Sort**:

Best case: O(n log n)

Average case: O(n log n)

Worst case: O(n^2) (when the smallest or largest element is always chosen as the pivot)

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

**Ans :** Quick Sort is generally preferred over Bubble Sort because the former has an average case time complexity of O(n log n), compared to the time complexity of O(n^2) of Bubble Sort. That basically means Quick Sort is more efficient on larger data.