**UNDERSTANDING SORTING ALGORITHMS**

1.Understand the Problem:

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

About four different sorting algorithms,

* + **Bubble Sort:** A simple sorting algorithm that repeatedly steps through the list, compares adjacent elements, and swaps them if they are in the wrong order. This gets repeated until the list is sorted.
  + **Insertion Sort:** A simple sorting algorithm that builds the final sorted array (or list) 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:** A divide-and-conquer algorithm that picks an element as a pivot, partitions the given array around the picked pivot, and then recursively applies the above steps to the sub-array of elements with smaller values and separately to the sub-array of elements with greater values.
  + **Merge Sort:** A divide-and-conquer algorithm that divides the input array into two halves, calls itself for the two halves, and then merges the two sorted halves.

4. Analysis:

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

**Bubble Sort:**

* Time complexity: O(n^2)
* Best-case scenario: O(n) (if the input is already sorted)
* Average-case scenario: O(n^2)
* Worst-case scenario: O(n^2)

**Quick Sort:**

* Time complexity: O(n log n) on average
* Best-case scenario: O(n log n)
* Average-case scenario: O(n log n)
* Worst-case scenario: O(n^2) (this can be avoided with proper pivot selection)
* Discuss why Quicksort is generally preferred over Bubble Sort..

Bubble sort has a simple implementation and it is not that much efficient for large datasets.Quick Sort, on the other hand, is a more efficient and performs well with large datasets.And also, Quick Sort has an average-case time complexity of O(n log n), making it much faster than Bubble Sort, which has a time complexity of O(n^2).