Que1. Explain different sorting algorithms (Bubble Sort, Insertion Sort, Quick Sort, Merge Sort)?

Ans: Bubble Sort:

* Traverse from left and compare adjacent elements and the higher one is placed at right side.
* In this way, the largest element is moved to the rightmost end at first.
* This process is then continued to find the second largest and place it and so on until the data is sorted.

Insertion Sort:

* Start with the second element the first one is assumed to be sorted.
* Compare the current position with the previous position
* If the current element is smaller than previous element then swap, else do nothing
* Traverse till all the elements are at their correct place

Quick Sort:

* It is a divide and conquer algorithm
* Consider a pivot element from the array (generally the middle element is taken)
* Iterate the array form start and also from end
* Find the element larger than pivot from start and less than pivot form end
* When found those elements swap
* Recursively perform this operation till the array is sorted

Merge Sort:

* Recursively Divide the array into two halves
* Sort each half
* Merge the sorted half

Que2. Compare the performance (time complexity) of Bubble Sort and Quick Sort?

Ans: Bubble Sort:

* Best-case: O(n) when the array is already sorted.
* Average-case: O(n^2) when the elements are placed at random indexes.
* Worst-case: O(n^2) when the elements are sorted in reverse order.

Quick Sort:

* Best-case: O(n logn)
* Average-case: On average quick sort always take O(n logn)
* Worst-case: O(n^2) always selecting the smallest or the largest element as pivot, this can be avoided by using middle element as pivot.

Que3. Discuss why Quick Sort is generally preferred over Bubble Sort?

Ans: Quick sort is generally preferred over bubble sort is because the time complexity of quick sort is less than bubble sort, which may don’t have impact on smaller sets of data but on larger data quicksort is more reliable and efficient.