DS- Assignment

Auchitya Gautam
IT
11912050

Q1. Insertion Sort

Algo :-

```
insertion sort (int arr [], size)
{
  for i → size
    j = arr [i];
    K = j - 1;
  while K >= 0 && arr [K] > j
    arr (K+1) = arr (k)
      K--;
  end while
  arr [K+1] = j
  end for
}
```

→ Since Insertion Sort is modifying the original array by inserting the lower element at the right place in the original array. Thus, it does not require any extra space. Hence, it is an "In-Place Sorting" Algorithm.

∴ **Space Complexity = O(1)**

→ 2 basic operation takes place in the algo :-
i) comparision
ii) swapping
considering both these operations cost the same.
In Best Case i.e. the array is already sorted
This algorithm only compares 'n' elements.

∴ Time Complexity = $O(n)$

Q2. → Quick Sort :-
→ Divide & conquer algorithm.
→ Time complexity
1) Worst Case $T(n) = O(n^2)$

2) Avg. Case $T(n) = O(n \log n)$

3) Best Case $T(n) = O(n \log n)$

→ Bubble Sort
Time Complexity.

For n elements (n-1) comparison are done :-

∴ $T(n) = 1+2+3 \dots (n-1)$
$\Rightarrow \frac{n^2-n}{2}$
$\Rightarrow O(n^2)$

→ Both quick sort & Bubble sort algorithm are "In-place" algorithm.

→ Bubble sort is efficient for small size array.

- Time complexity for merge sort → $O(n \log n)$
- Time complexity for Insertion sort → $\theta(n^2)$