DS - Assignment

Ansh Tyagi
IT, 11912079

Q1 Insertion Sort

Algo :-

```
insertionSort (int arr[], size)
{
        for i → size
            j = arr[i];
            k = j - 1;
            while k >= 0 && arr[k] > j
                arr(k+1) = arr(k)
                k--;
            endwhile
            arr[k+1] = j
        endfor
}
```

• Since, Insertion Sort is modifying the original array by inserting the lower element at the right place in the original array only. 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)

A2 → **Quick Sort** :-

- Divide & Conquer algorithm
- Time Complexity

1) **Worst Case**,

By master Theorem,

$$\boxed{T(n) = O(n^2)}$$

2) Avg Case,

$$\boxed{T(n) = O(n \log n)}$$

3) Best Case

$$\boxed{T(n) = O(n \log n)}$$

→ **Bubble Sort**

- Time Complexity

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

$$\therefore T(n) = 1 + 2 + \dots + (n-1)$$

$$\Rightarrow \frac{n(n-1)}{2} \Rightarrow \frac{n^2 - n}{2}$$

$$\Rightarrow \boxed{O(n^2)}$$

→ Both, Quick Sort & Bubble sort algorithms are "In-Place" Algorithm

→ Bubble sort is efficient for small size arrays

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