DS - ASSIGNMENT

NILESH GUPTA
IT, 11912015

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 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

In Best case i.e, the array is already sorted This algorithm only compares 'n' elements,

$$\therefore \text{Time Complexity} = O(n)$$

Quick Sort :-

- Divide & Conquer algorithm
- Time complexity

1) Worst case,

By master Theorem,

$$T(n) = O(n^2)$$

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) comparisions are done :-

∴ $T(n) = 1+2+3+\cdots+(n-1)$

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

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

⇒ 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 → $O(n^2)$