DS - Assignment

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 take 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)

Quick sort

* Divide & conquer algorithm
* Time complexity

Worst case:

By master theorem

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

Avg case,

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

Best case:

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

Bubble sort

* Time complexity

for n elements $(n-1)$ comparisons are done

$\therefore T(n) = 1 + 2 + 3 \ldots + (n-1)$

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

$\Rightarrow O(n^2)$

→ Both, Quick & Bubble sort algorithms are "In-pla
Algorithm.

→ Bubble sort is efficient for small size arrays

{ Time complexity merge sort — $O(n \log n)$
Time complexity Insertion sort — $O(n^2)$ }