DS - Assignment

Q1. Insertion Sort

Algorithm

```
insert(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
}
```

Insertion does not require any extra space. It only modifies the original array by moving the smallest element to front.

Hence space complexity O(1).

Insertion sort sorts the array by comparing & then swapping.

In best case, ~~time~~ time complexity = O(n)

2. Quick Sort

Time complexity :-

Worst Case : $O(n^2)$

Avg Case : $O(n \log n)$

Best Case : $O(n \log n)$

3. Bubble Sort :

For n elements $(n+1)$ comparisons are made

$\therefore$ Time complexity $= O(n^2)$

Note : Quick Sort & Bubble Sort are "in-place" algorithms.

Bubble sort is efficient for arrays with small size.