Name - Tushar Singh
Branch - IT
Roll no - 1191203

Q1 array - arr[] = {1,2,3,4,7} (sorted)
Insertion sort →

```
for (j=2; j<n; j++)
{
  Key = arr[j]
  i = j-1
  while (i>0 && Key < arr[i])
{
  arr[i+1] = arr[i];
}
  arr[i+1] = Key
}
```

Complexity analysis

	0	1	2	3	4
		1	2	3	4
i			1	2	3
j			2	3	4
Key			2	3	4
Comp			1	1	1
Moved			1	1	1

$P(n) = 2+2+2 \cdots n = 2(n)$

time complexity = $O(n)$

The complexity can be reduced by

1) Binary search - By using a sorted array & using binary search can reduced the complexity to O(n)

2) By using linked list :- complexity of insertion become const & the overall complexity becomes O(n)

Q2 Quick sort algorithm

```
Quick sort (arr, p, q)
  if (p < q)
  r = partition (arr, p, q)
Quick_sort (arr, p, r-1)
Quick sort (arr, r+1, q)
partition
  n = arr[q]
  i = p-1
  for (j = p to q-1)
  if (arr[j] < n)
  swap (arr[j], arr[++i])
arr[++i] = n
return i
```

The complexity of best case of quick sort = O(n log n)

The complexity of quick sort is $P(n) = O(n^2)$

Bubble sort

```
for (i = 0 to size)
  for (j = 0 to size - 1)
  if ( a[j] > a[j+1])
     temp = a[j]
     a[j] = a[j+1]
     a[j+1] = a[j]
```

The complexity of bubble sort

best $= O(n^2)$

worst $= O(n^2)$