Tutorial - 3

Name - Amandeep Singh
Section - G
Roll No. - 53
Uni Roll No. - 2016623
Course - B. Tech. CSE
Semester - 4

Q1

Ans 1 Pseudofunction for linear Search :-

```
int linear ( int * arr[] , int n , int key)
{
    for (int i= 0; i<n; i++)
        {
            if ( arr[i] == key)
            return i;
        }
    return -1;
}
```

Q2

Ans 2 Insertion Sort :-

```
void insertion ( int arr[] , int n)
{
    for (int i = 1 ; i<n ; i++)
    {
        int key = arr[i];
        int j = i-1;
        while ( j>=0 && arr[j] > key)
        {
            arr[j+1] = arr[j];
            j--;
        }
    arr[j+1] = key;
    }
}
```

→ Insertion Sort is called online sort as if an element comes in an array it is automatically inserted at its correct position.

Q3

Ans 3 Average case complexity of Sorting Algos

- Bubble = $O(n^2)$
- Insertion = $O(n^2)$
- Selection = $O(n^2)$
- Merge = $O(n \log n)$
- Quick = $O(n \log n)$
- Heap = $O(n \log n)$

Q4

Ans 4

	Stable (appears in same order)	Inplace
Bubble	✓	O(1) ✓
Selection	✗	✓
Insertion	✓	✓
Merge	✓	✓
Quick	✗	✗
Heap	✗	✗

Q5

Ans 5 binary Search :-

```
int start = 0
int end = size - 1
while ( start <= end)
{
  int mid [start + (end - start)] / 2;
  if (key == arr [mid])
      return mid;
  else if (key < arr [mid])
      end = mid - 1;
```

```
else
    start = mid + 1;
}
return -1;
```

→ Time complexity = $O(\log n)$

→ Space complexity = $O(1)$

Linear Search :-

→ T.C. = $O(n)$

→ S.C. = $O(1)$

Q6

Ans 6 Reoccurence relation of Binary Search :-

$$T(n) = T(n/2) + 1$$

Q8

Ans 8 Quick sort is the best sorting algo in practical use as it follows the locality of reference & also its best case time complexity is $O(n \log n)$

Q9

Ans 9 No. of inversions : It tells us how far is the array is from being sorted.

if $a[i] > a[j]$ & $i < j$

→ 7 21 31 8 10 1 20 6 48

no. of inversions : 4 + 7 + 7 + 4 + 4 + 3 + 2

=> 31

Q10

Ans 10 Quick sort will give :-

* Best case complexity : when array is totally unsorted.
* Worst complexity : when array is sorted or reverse sorted.

Q11

Ans 11 Recurrence Relation of :-

Merge Sort	Quick Sort
Best, Worst : $2T(n/2) + \Theta(n)$	$T(n) = T(k) + T(n-k-1) + \Theta(n)$
	$T(n) = T(n-1) + \Theta(n)$

Similarity : Both are of type Divide & Conquer

Difference : Worst case complexity of merge sort is $O(n \log n)$ where as of Quick Sort is $O(n^2)$

Q 13

Ans 13 Optimised Bubble Sort :-

```
for (int i=0 ; i<n ; i++)
  {
    swap = false ;
    for (j=0 ; j<n-i-1 ; j++)
    {
      if (arr[j] < arr[j+1])
      {
        swap (arr[j], arr[j+1]);
        swap = true;
      }
    }
  }
```

Q 14

Ans 14 → In such case, Merge Sort would be efficient as it is an external sorting algorithm i.e. data is divided into chunks & then sorted using merge sort.

→ Sorted data is dumped into files.

* Internal Sorting :- It is a type of sort in which whole sorting takes place in main memory of computer.