Practical – 10

Sorting techniques

Sorting refers to arranging data in a particular format. Sorting algorithm specifies the way to arrange data in a particular order. Most common orders are in numerical or lexicographical order.
Sorting is the process of arranging the elements of an array so that they can be placed either in ascending or descending order. For example, consider an array A = {A1, A2, A3, A4, ?? An }, the array is called to be in ascending order if element of A are arranged like A1 > A2 > A3 > A4 > A5 > ? > An .

Consider an array;

int A[10] = { 5, 4, 10, 2, 30, 45, 34, 14, 18, 9)

The Array sorted in ascending order will be given as;

A[] = { 2, 4, 5, 9, 10, 14, 18, 30, 34, 45 }

There are many techniques by using which, sorting can be performed. In this section of the tutorial, we will discuss each method in detail.

Bubble sort works on the repeatedly swapping of adjacent elements until they are not in the intended order. It is called bubble sort because the movement of array elements is just like the movement of air bubbles in the water. Bubbles in water rise up to the surface; similarly, the array elements in bubble sort move to the end in each iteration.

Bubble sort alogrithm

```
begin BubbleSort(arr)
  for all array elements
    if arr[i] > arr[i+1]
      swap(arr[i], arr[i+1])
    end if
  end for
  return arr
end BubbleSort
```

Quick Sort

Quick sort partitions an array and then calls itself recursively twice to sort the two resulting sub arrays.

```
FLAG ← true
IF   LB < UB
Then
 I ← LB
 J ← UB + 1
 KEY ← K[LB]
```

```
Repeat While FLAG = true
  I ← I+1
  Repeat While K[I] < KEY
   I ← I + 1
  J ← J – 1
  Repeat While K[J] > KEY
   J ← J – 1
  IF   I<J
  Then K[I] ←---→ K[J]
  Else FLAG ← FALSE

 K[LB] ←---→ K[J]
```

Insertion sort

- **Step 1 -** If the element is the first element, assume that it is already sorted. Return 1.
- **Step2 -** Pick the next element, and store it separately in a **key.**
- **Step3 -** Now, compare the **key** with all elements in the sorted array.
- **Step 4 -** If the element in the sorted array is smaller than the current element, then move to the next element. Else, shift greater elements in the array towards the right.
- **Step 5 -** Insert the value.
- **Step 6 -** Repeat until the array is sorted.

Exercise

1. Write a program to implement bubble sort.
2. Write a program to implement quick sort.
3. Write a program to implement insertion sort