

Lab-11

Counting Sort

Objective:

This lab will provide you with the inside knowledge of another very important type of Sorting, i.e. Counting Sort.

Activity Outcomes:

- implement the sorting technique without comparison
- **Useful Concepts**
 - Counting sort assumes that the numbers to be sorted are in the range 1 to k where k is small.
 - The basic idea is to determine the rank of each number in a final sorted array.
 - Recall that the rank of an item is the number of elements that are less than or equal to it.
 - Once we know the ranks, we simply copy numbers to their final position in an output array.

Activity 1:

Create ranking array 'C'

- First find the maximum number in the original array 'A'

```
public static int findMax(int[] A)
{
    int max = Integer.MIN_VALUE;

    for (int i=0;i<A.length;i++)
    {
        if(A[i]>max)
            max = A[i];
    }

    return max;
}
```

Activity 2:

Create the ranking array

```
public static void createRanking(int[] A)
{
    //array size = maximum element in the array A
    C = new int[findMax(A)+1];

    //initialize the array with zeros
    for(int i=0;i<C.length;i++)
        C[i] = 0;

    //start ranking
    for(int j=0;j<A.length;j++)
        C[A[j]]++;

    System.out.print("ranking -> ");
    for(int k=0;k<C.length;k++)
        System.out.print(C[k]+",");

    System.out.println();
}
```

```

        System.out.print("extra spaces -> ");
        //create extra spaces as C.length < A.length
        for(int i=2;i<C.length;i++)
            C[i] = C[i] + C[i-1];

        for(int k=0;k<C.length;k++)
            System.out.print(C[k]+",");
    }

```

Activity 3:

Copy the array A into B in a sorted manner using C array

```

public static int[] sortedArray(int[] A)
{
    B = new int[A.length];

    for(int i=0;i<A.length;i++)
    {
        B[C[A[i]]-1] = A[i];
        C[A[i]]--;
    }
    return B;
}

```

Lab Task

Write a program for the Radix Sort