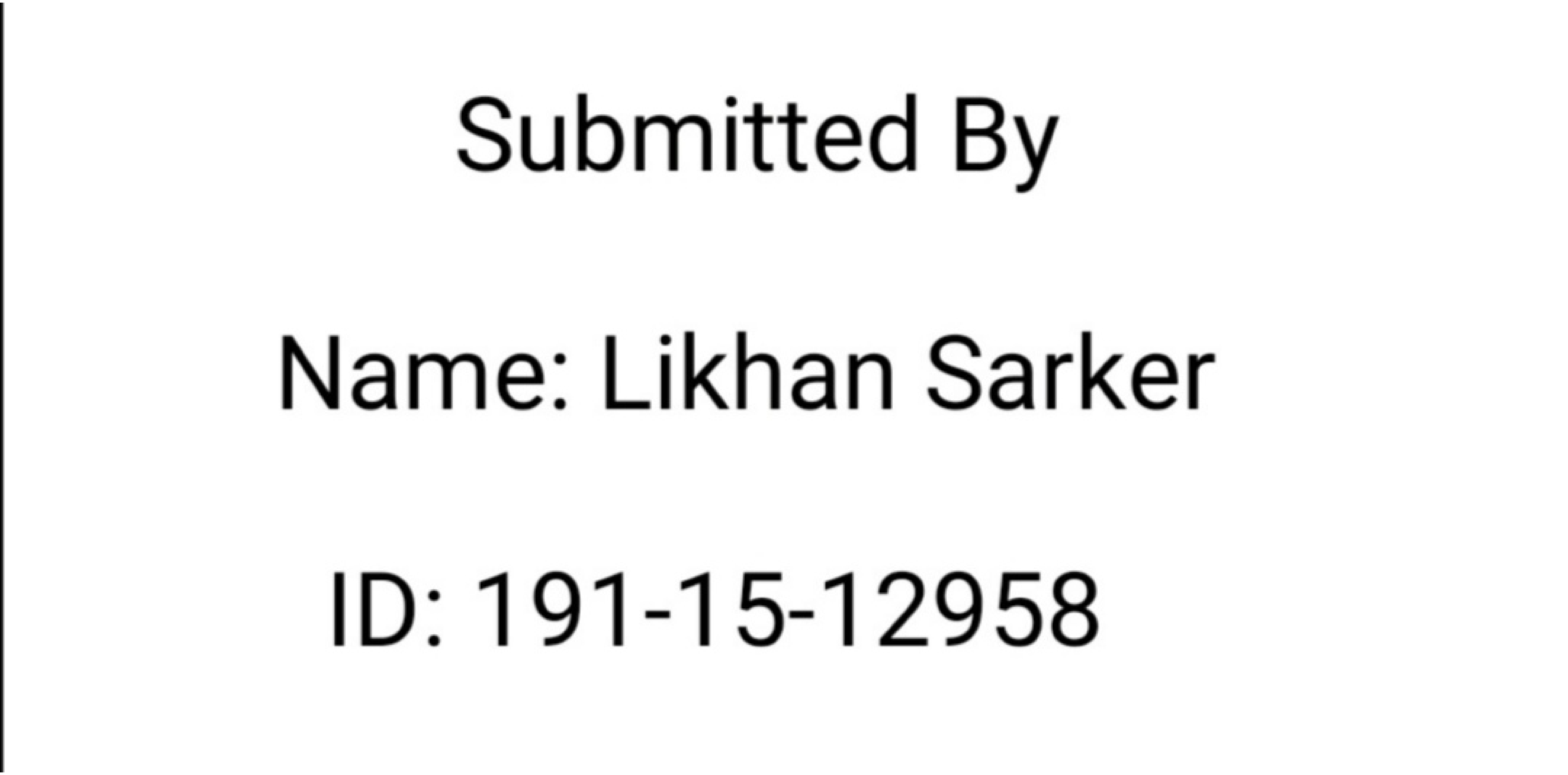

Submitted By

Name: Likhan Sarker

ID: 191-15-12958

① Bubble Sort

```
Package
Com. Company;

     Class BubbleSort
     {
        void bubbleSort (int a[])
         {
           int n=a.length;
            for (int i=0; i<n-1; i++)
              for(int j=0; j<n-i; j++)
                if (a[j] > a[j+1])
                {
                   int flag=a[j];
                    a[j]=a[j+1];
                    a[j+1]=flag;
                  }

         }
         void PrintArray(int x[])
{
  int n = x.length;
  for (int i = 0 ; i<n ; ++i)
     System.out.Print (x [i] + " ");
   System.out.Println ();
 }
 Public Static void main (String args [])
 {
   Bubble Sort ob = new BubbleSort();
   int array [] = {12, 8, 7, 5, 2};
   ob. bubble Sort (array);
   System.out. Print ln ("Sorted Array");
   ob. Print Array (array);
  }
 }
```

Bubble Sort Algorithm:

Worst Case Performance $O(n^2)$

Best case Performance $O(n)$. Average case Performance $O(n^2)$

④ Linear Search

```
Package
Com.Company;
   Class LinearSearch
   {
    Public Static int Search(int a[], int x]
    {
     int n=a.length;
     for (int i=0; i<n; i++)
     {
      if (a[i]==x)
       return i;
     }
      return -1;
    }
    Public Static void main (String args[])
    {
     int x[]={5,25,90,29};
     int v=90;
int result = Search(x, v);
if (result == -1)
    System.out.Print("Element is not
                         available in array");
    else
    System.out.Print("Element found at
                         index " + result);
  }
}
```

Linear Search Algorithm ;

Worst Case performance $O(n)$

Best " " $O(1)$

Average " " $O(n)$

(41) Insertion Sort

```
Package
Com. Company;
      import Java.Util.Arrays;
      class Insertion Sort{
        void insertionSort(int array[]){
         int size=array.length;
         for(int step =1; step <size ; step++){
          int key =array [step];
          int j= step -1;
         while (j>= 0 && key<array[j]){
          array[j+1]= array[j];
           --j;
   }
```

```
  array[j+1]=key;
  }
}
Public Static void main(String args[]){
    int[] data ={9,5,1,4;3};
    InsertionSort is = new InsertionSort();
    is.insertion Sort (data);
    System.out.Println("Sorted Array in
                         Ascending order:");
    System.out.Println(Arrays.String (data
  }
}
```

Insertion Sort Algorithm:

Worst case performance O(n2)

Best " " O(n)

Average " " O(n2)

14 Selection Sort

```
Package
Com. Company;
 import Java, util, Scanner;
 {
 Public class selection
 {
  Public Static void main(String args[])
  {
   int size, i, J, temp;
    int arr[] = new int[50];
   Scanner Scan = new Scan(System.in);

   System . out.Print("Enter Array Size");
   Size = Scan. nextInt();
  System . out. Print ("Enter Array Element:");
  ~~size~~ for(i=0; i<size; i++)
```

```
{
 arr[i] = scan.nextInt();
}
System.out.Print("Sorting Array using selection
                        Sort Technique..\n");
for (i=0; i<size; i++)
{
 for (j= i+1; j<size; j++)
  {
   if (arr[i] > arr[j]
   {
    temp=arr[i];
     arr[i] = arr[j]);
     arr[j] = temp;
   }
  }
}
```

```
System.out.Print("Now the Array after
                        sorting in :\n");
 for (i=0; i<size ; i++)
 {
   System.out.Print(arr[i]+" ");
   }
  }
 }
```

Selection Sort Algorithm :

Worst Case Performance $O(n^2)$

Best " " $O(n^2)$

Average " " $O(n^2)$

(v) Binary Search

```
Class Binary Search Example{
Public Static void binary(int arr[],
            int first, int last, int key){
 int mid =(first + last)/2;
 while (first <=last){
 if (arr[mid +1;
 } elseif (arr[mid] == key){
 System.out.Println("Element isfound at
                    index:"+mid);
 break ;
 }else{
      last =mid-1;
 }
 mid= (first +last)/2;
}
```

```
  if (first>last){
     System.out.Println("Element is notfound!");
   }
 }
Public Static void main (String args[]){
     int arr[]={10,20,30,40,50};
     int key = 30;
     int last =arr.length-1;
      binary Search (arr,0,last,key);
   }
 }
```

Binary Search Algorithm:

Worst case performance. O(log n)

Best 〃 〃 O (1)

Average 〃 〃 O (logn)