Engineering Animuthyam

Java DSA – Course. Part - 45

Binary Search

Course Link:
https://www.youtube.com/playlist?list=PLjzLBp9HHZWhVXBSPS1VqxXXDoVk07gd9

Website Link:
https://www.vigneshreddyjulakanti.in/

Hello machas, Bagunnara

[1,2,2,2,2,3]

Find the index of right most 2;

```java
class HelloWorld {
    public static int rightMost(int arr[],int target
        ){
        int l =0;
        int r = arr.length - 1;
        while(l<=r){
            int m = l + (r-l)/2;
            if(arr[m] > target){
                r = m - 1;
            }else{
                l = m + 1;
            }
        }
        return r;
    }
    public static void main(String[] args) {
        int arr[]={1,2,2,2,2,3};
        int target = 2;
        System.out.println(rightMost(arr,target));
    }
}
```

Find the index of left most 2;

```
class HelloWorld {
    public static int LeftMost(int arr[],int target
        ){
        int l =0;
        int r = arr.length - 1;
        while(l<=r){
            int m = l + (r-l)/2;
            if(arr[m] >= target){
                r = m - 1;
            }else{
                l = m + 1;
            }
        }
        return l;
    }
    public static void main(String[] args) {
        int arr[]={1,2,2,2,2,3};
        int target = 2;
        System.out.println(LeftMost(arr,target));
    }
}
```

34. Find First and Last Position of Element in Sorted Array

https://leetcode.com/problems/find-first-and-last-position-of-element-in-sorted-array/

```
class Solution {
    public int LeftMost(int arr[],int target){
        int l =0;
        int r = arr.length - 1;
        while(l<=r){
            int m = l + (r-l)/2;
            if(arr[m] >= target){
                r = m - 1;
            }else{
                l = m + 1;
            }
        }
        if(l >= arr.length){
            return -1;
        }
        if(arr[l] != target){
            return -1;
        }
        return l;
    }
    public int RightMost(int arr[],int target){
        int l =0;
        int r = arr.length - 1;
        while(l<=r){
```

```
        while(l<=r){
            int m = l + (r-l)/2;
            if(arr[m] > target){
                r = m - 1;
            }else{
                l = m + 1;
            }
        }
        if(r < 0){
            return -1;
        }
        if(arr[r] != target){
            return -1;
        }
        return r;
    }
    public int[] searchRange(int[] nums, int target) {
        int lm = LeftMost(nums,target);
        int rm = RightMost(nums,target);
        int ans[]={lm,rm};
        return ans;

    }
}
```