Lecture-13

Binary Search Questions

Code S. Q.

First and Last Position of an Element In Sorted Array

https://www.codingninjas.com/studio/problems/first-and-last-position-of-an-element-in-sorted-array_1082549?source=youtube&campaign=love_babbar_codestudio2

One solution (wihtout Binary Search):
Complexity: O(n)

```
#include <bits/stdc++.h>
pair<int, int> firstAndLastPosition(vector<int>& arr, int n, int k)
{
pair<int, int> vp = {-1, -1}; // Initialize to invalid values
// Write your code here
for (int i = 0; i < n; i++) {
if (arr[i] == k) {
vp.first = i;
break; // Stop once first occurrence is found
}
}
for (int i = vp.first; i < n; i++) {
if (arr[i] == k) {
vp.second = i;
}
}
return vp;
}
```

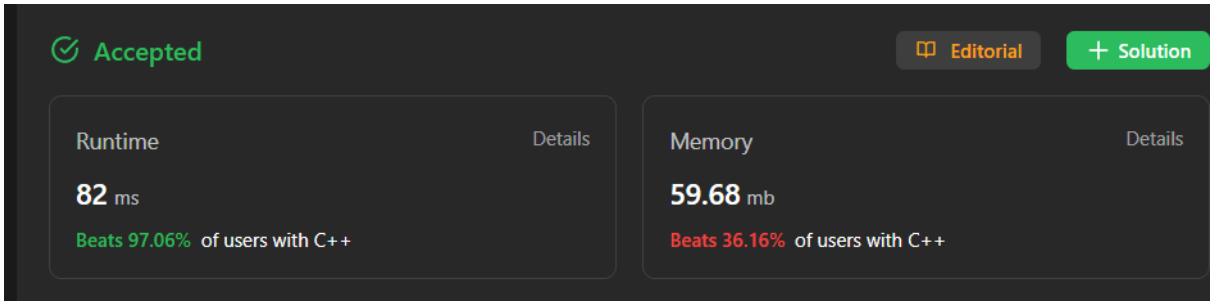
Optimized Solution(using Binary Search)
Complexity: O(log(n))

```
#include <bits/stdc++.h>
int firstOccurence(vector<int>& arr, int n, int key){
int start = 0, end=n-1;
```

```
int mid = start + (end-start)/2;
int ans=-1;
while(start<=end){
if(arr[mid]==key){
ans = mid;
end = mid-1;
}
else if(arr[mid]>key){
end = mid-1;
}
else{
start=mid+1;
}
mid = start + (end-start)/2;
}
return ans;
}
int lastOccurence(vector<int>& arr, int n, int key){
int start=0;
int end=n-1;
int mid = start + (end-start)/2;
int ans=-1;
while(start<=end){
if(arr[mid]==key){
ans = mid;
start=mid+1;
}
else if(arr[mid]>key){
end = mid-1;
}
else{
start = mid+1;
}
mid = start + (end-start)/2;
}
return ans;
}
pair<int, int> firstAndLastPosition(vector<int>& arr, int n, int k)
{
pair<int, int> vp;
vp.first = firstOccurence(arr, n, k);
vp.second = lastOccurence(arr, n, k);
return vp;
}
```

Leetcode 852.

https://leetcode.com/problems/peak-index-in-a-mountain-array/
852. Peak Index in a Mountain Array



- Important Point: We only have to move the loop till start<end and not start<=end otherwise TLE

```
class Solution {
public:
int peakIndexInMountainArray(vector<int>& arr) {
int start = 0, end = arr.size()-1;
int mid = start + (end-start)/2;
while(start<end){
if(arr[mid]<arr[mid+1]){
start = mid + 1;
}
else{
end = mid;
}
mid = start + (end-start)/2;
}
return mid;
}
};
```