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

Given an array of integers `nums` sorted in ascending order, find the starting and ending position of a given `target` value.

If `target` is not found in the array, return `[-1, -1]`.

You must write an algorithm with `O(log n)` runtime complexity.

Example 1:

```
Input: nums = [5,7,7,8,8,10], target = 8
Output: [3,4]
```

Example 2:

```
Input: nums = [5,7,7,8,8,10], target = 6
Output: [-1,-1]
```

Example 3:

```
Input: nums = [], target = 0
Output: [-1,-1]
```

Constraints:

- `0 <= nums.length <= 10<sup>5</sup>`
- `-10<sup>9</sup> <= nums[i] <= 10<sup>9</sup>`
- `nums` is a non-decreasing array.
- `-10<sup>9</sup> <= target <= 10<sup>9</sup>`

```python
class Solution:
    def searchRange(self, nums: List[int], target: int) -> List[int]:
        if len(nums)==0:
            return [-1,-1]
        n = len(nums)-1

        lo = 0
        hi = n
        while lo<=hi:
```

```
            mid = (lo+hi)//2
            if nums[mid]==target:
                L = self.leftOccurance(nums,0,mid,target)
                R = self.rightOccurance(nums,mid+1,n,target)
                return [L,R]
            elif nums[mid]>target:
                hi = mid-1
            else:
                lo = mid+1
        return [-1,-1]


    def leftOccurance(self,nums,lo,hi,target):
        while lo<=hi:
            mid = (lo+hi)//2
            if nums[mid]==target:
                hi = mid-1
            elif nums[mid]>target:
                hi = mid-1
            else:
                lo = mid+1
        return lo

    def rightOccurance(self,nums,lo,hi,target):

        while lo<=hi:
            mid = (lo+hi)//2
            if nums[mid]<=target:
                lo = mid+1
            else:
                hi = mid-1
        return lo-1
```

```
class Solution:
    def searchRange(self, nums: List[int], target: int) -> List[int]:
        i = self.firstOccurrence(nums,target)
        j = self.lastOccurrence(nums,target)
        return i,j



    def firstOccurrence(self,arr,x):
        lo = 0
```

```
        hi = len(arr)-1
        res = -1
        while lo<=hi:
            mid = lo + (hi-lo)//2
            if arr[mid]==x:
                hi = mid-1
                res = mid
            elif arr[mid]>x:
                hi = mid-1
            else:
                lo = mid+1
        return res

    def lastOccurrence(self,arr,x):
        lo = 0
        hi = len(arr)-1
        res = -1
        while lo<=hi:
            mid = lo + (hi-lo)//2
            if arr[mid]==x:
                lo = mid+1
                res = mid
            elif arr[mid]>x:
                hi = mid-1
            else:
                lo = mid+1
        return res
```