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]

```
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
```