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

Given an array of integers nums sorted in non-decreasing order, find the starting and

ending position of a given target value.

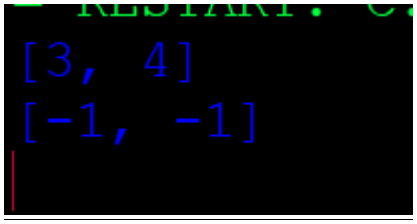
If target is not found in the array, return [-1, -1].

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

Code:

```
def searchRange(nums, target):
    def findFirst(nums, target):
        left, right = 0, len(nums) - 1
        first = -1
        while left <= right:
            mid = (left + right) // 2
            if nums[mid] >= target:
                right = mid - 1
            else:
                left = mid + 1
            if nums[mid] == target:
                first = mid
        return first

    def findLast(nums, target):
        left, right = 0, len(nums) - 1
        last = -1
        while left <= right:
            mid = (left + right) // 2
            if nums[mid] <= target:
                left = mid + 1
            else:
                right = mid - 1
            if nums[mid] == target:
                last = mid
        return last

    first = findFirst(nums, target)
    last = findLast(nums, target)

    return [first, last]
nums1 = [5, 7, 7, 8, 8, 10]
target1 = 8
print(searchRange(nums1, target1))

nums2 = [5, 7, 7, 8, 8, 10]
target2 = 6
print(searchRange(nums2, target2))
```

Output:



Time Complexity:

- T(n)= O(logn)