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.

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]

AIM: To Find First and Last Position of Element in Sorted Array

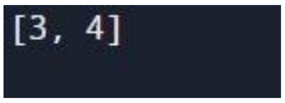
PROGRAM:

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

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

    left_index = findLeft(nums, target)
    right_index = findRight(nums, target)
    if left_index <= right_index:
        return [left_index, right_index]
    else:
        return [-1, -1]


nums1 = [5, 7, 7, 8, 8, 10]
target1 = 8
print(searchRange(nums1, target1))
```

OUTPUT: 

```
[3, 4]
```

TIME COMPLEXITY: O(log n)