

```

import Foundation
/// 剑指 Offer 53 - I. 在排序数组中查找数字 I
/// 统计一个数字在排序数组中出现的次数。
///
/// 示例 1:
/// 输入: nums = [5,7,7,8,8,10], target = 8
/// 输出: 2
class Solution {
    func helper(_ nums: [Int], _ target: Int, _ lower: Bool) -> Int {
        var start = 0
        var end = nums.count - 1
        while(start <= end) {
            /// 取区间中点, 向下取
            let mid = start + (end-start) >> 2
            if (nums[mid] > target) {
                /// 如果当前的数值大于目标值, 则说明在mid左半边
                end = mid - 1
            } else if (nums[mid] < target) {
                /// 如果当前的数值小于目标值, 则说明在mid右半边
                start = mid + 1
            } else {
                if lower {
                    /// 如果找最左边的位置, 则减小最右边的索引
                    end = mid - 1
                } else {
                    start = mid + 1
                }
            }
        }
        /// 找最左边返回最右边的数值
        return lower ? end : start
    }

    func search(_ nums: [Int], _ target: Int) -> Int {
        /// 按这样找到的是不包含目标数字的左右索引, 需要减1
        return helper(nums, target, false) - helper(nums, target, true) - 1
    }
}

let solution = Solution()
print(solution.search([5,7,7,8,8,10], 8))
print(solution.search([5,7,7,8,8,10], 6))

```