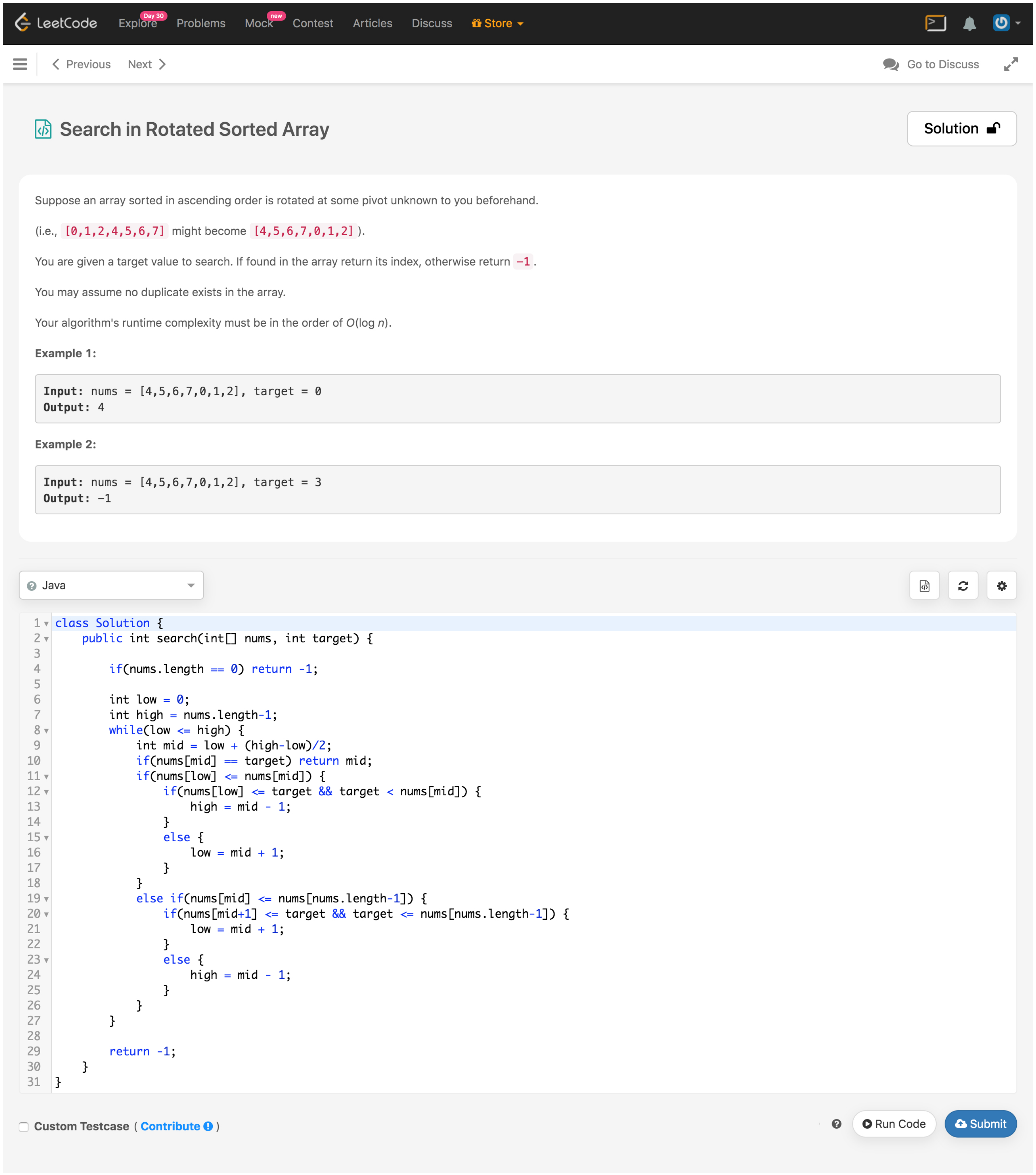

Search in Rotated Sorted Array

Suppose an array sorted in ascending order is rotated at some pivot unknown to you beforehand.

(i.e., `[0,1,2,4,5,6,7]` might become `[4,5,6,7,0,1,2]`).

You are given a target value to search. If found in the array return its index, otherwise return `-1`.

You may assume no duplicate exists in the array.

Your algorithm's runtime complexity must be in the order of *O*(log *n*).

Example 1:

```
Input: nums = [4,5,6,7,0,1,2], target = 0
Output: 4
```

Example 2:

```
Input: nums = [4,5,6,7,0,1,2], target = 3
Output: -1
```

```java
class Solution {
    public int search(int[] nums, int target) {

        if(nums.length == 0) return -1;

        int low = 0;
        int high = nums.length-1;
        while(low <= high) {
            int mid = low + (high-low)/2;
            if(nums[mid] == target) return mid;
            if(nums[low] <= nums[mid]) {
                if(nums[low] <= target && target < nums[mid]) {
                    high = mid - 1;
                }
                else {
                    low = mid + 1;
                }
            }
            else if(nums[mid] <= nums[nums.length-1]) {
                if(nums[mid+1] <= target && target <= nums[nums.length-1]) {
                    low = mid + 1;
                }
                else {
                    high = mid - 1;
                }
            }
        }

        return -1;
    }
}
```