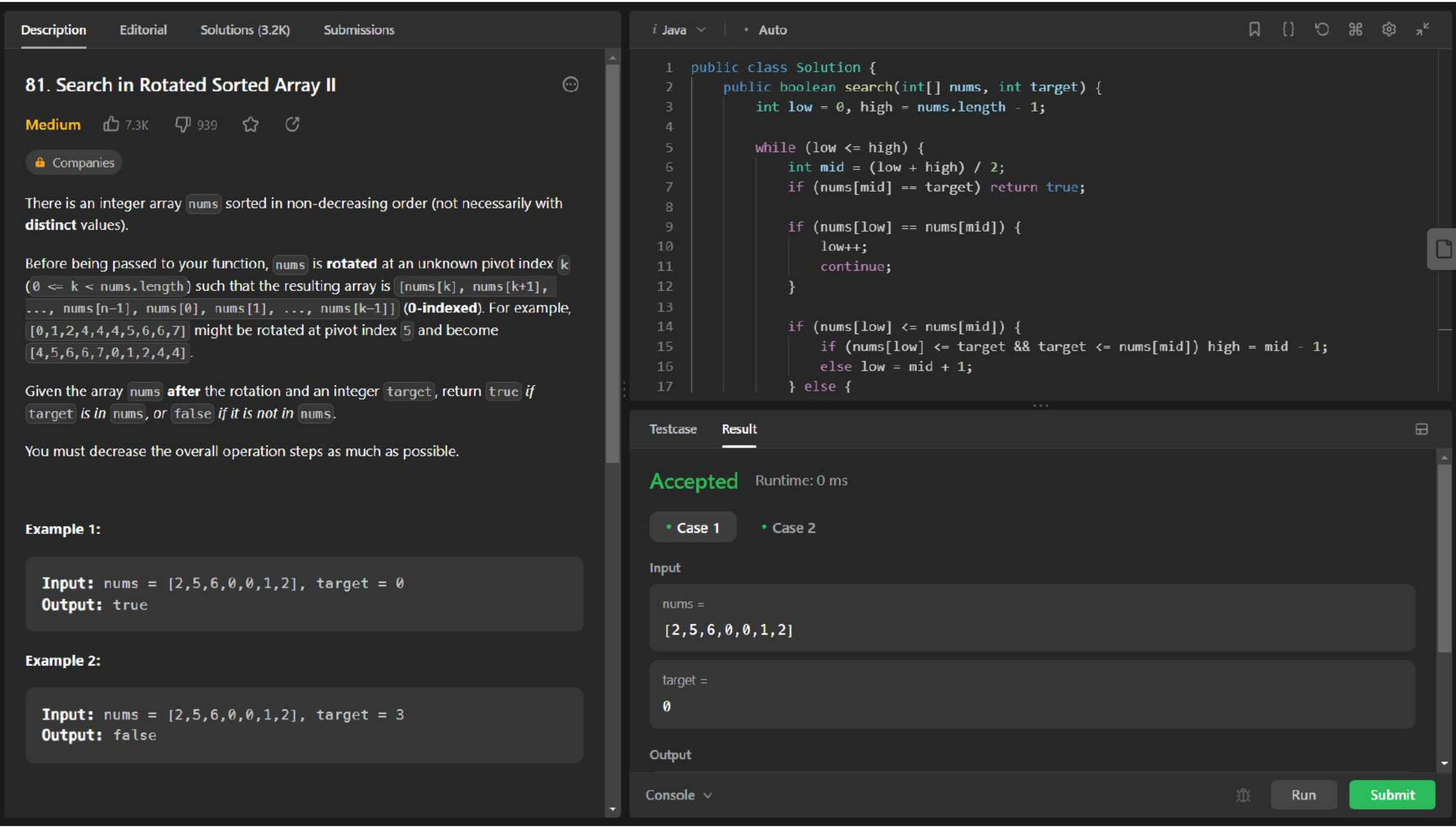


81. Search in Rotated Sorted Array II

There is an integer array `nums` sorted in non-decreasing order (not necessarily with **distinct** values).

Before being passed to your function, `nums` is **rotated** at an unknown pivot index `k` (`0 <= k < nums.length`) such that the resulting array is `[nums[k], nums[k+1], ..., nums[n-1], nums[0], nums[1], ..., nums[k-1]]` (**0-indexed**). For example, `[0,1,2,4,4,4,5,6,6,7]` might be rotated at pivot index `5` and become `[4,5,6,6,7,0,1,2,4,4]`.

Given the array `nums` **after** the rotation and an integer `target`, return `true` *if* `target` *is in* `nums`*, or* `false` *if it is not in* `nums`*.*

You must decrease the overall operation steps as much as possible.

Example 1:

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

Example 2:

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

```java
public class Solution {
    public boolean search(int[] nums, int target) {
        int low = 0, high = nums.length - 1;

        while (low <= high) {
            int mid = (low + high) / 2;
            if (nums[mid] == target) return true;

            if (nums[low] == nums[mid]) {
                low++;
                continue;
            }

            if (nums[low] <= nums[mid]) {
                if (nums[low] <= target && target <= nums[mid]) high = mid - 1;
                else low = mid + 1;
            } else {
```

Accepted Runtime: 0 ms

• Case 1 • Case 2

Input

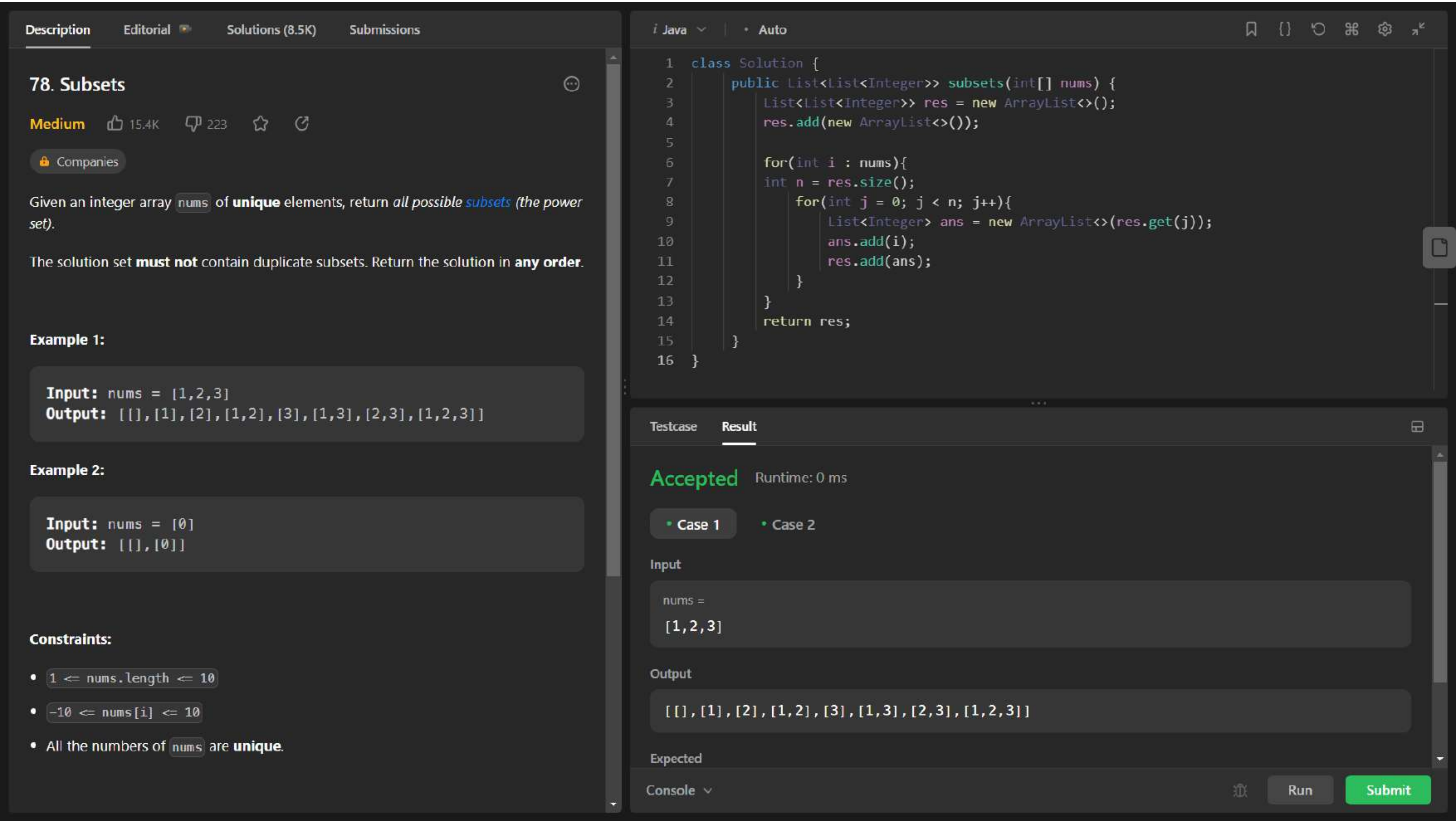
nums =

```
[2,5,6,0,0,1,2]
```

target =

```
0
```

Output



78. Subsets

Medium 15.4K 223

Companies

Given an integer array `nums` of **unique** elements, return *all possible subsets (the power set)*.

The solution set **must not** contain duplicate subsets. Return the solution in **any order**.

Example 1:

```
Input: nums = [1,2,3]
Output: [[],[1],[2],[1,2],[3],[1,3],[2,3],[1,2,3]]
```

Example 2:

```
Input: nums = [0]
Output: [[],[0]]
```

Constraints:

- `1 <= nums.length <= 10`
- `-10 <= nums[i] <= 10`
- All the numbers of `nums` are **unique**.

```java
class Solution {
    public List<List<Integer>> subsets(int[] nums) {
        List<List<Integer>> res = new ArrayList<>();
        res.add(new ArrayList<>());

        for(int i : nums){
        int n = res.size();
            for(int j = 0; j < n; j++){
                List<Integer> ans = new ArrayList<>(res.get(j));
                ans.add(i);
                res.add(ans);
            }
        }
        return res;
    }
}
```

Accepted Runtime: 0 ms

Case 1 Case 2

Input

nums =

```
[1,2,3]
```

Output

```
[[],[1],[2],[1,2],[3],[1,3],[2,3],[1,2,3]]
```

Expected