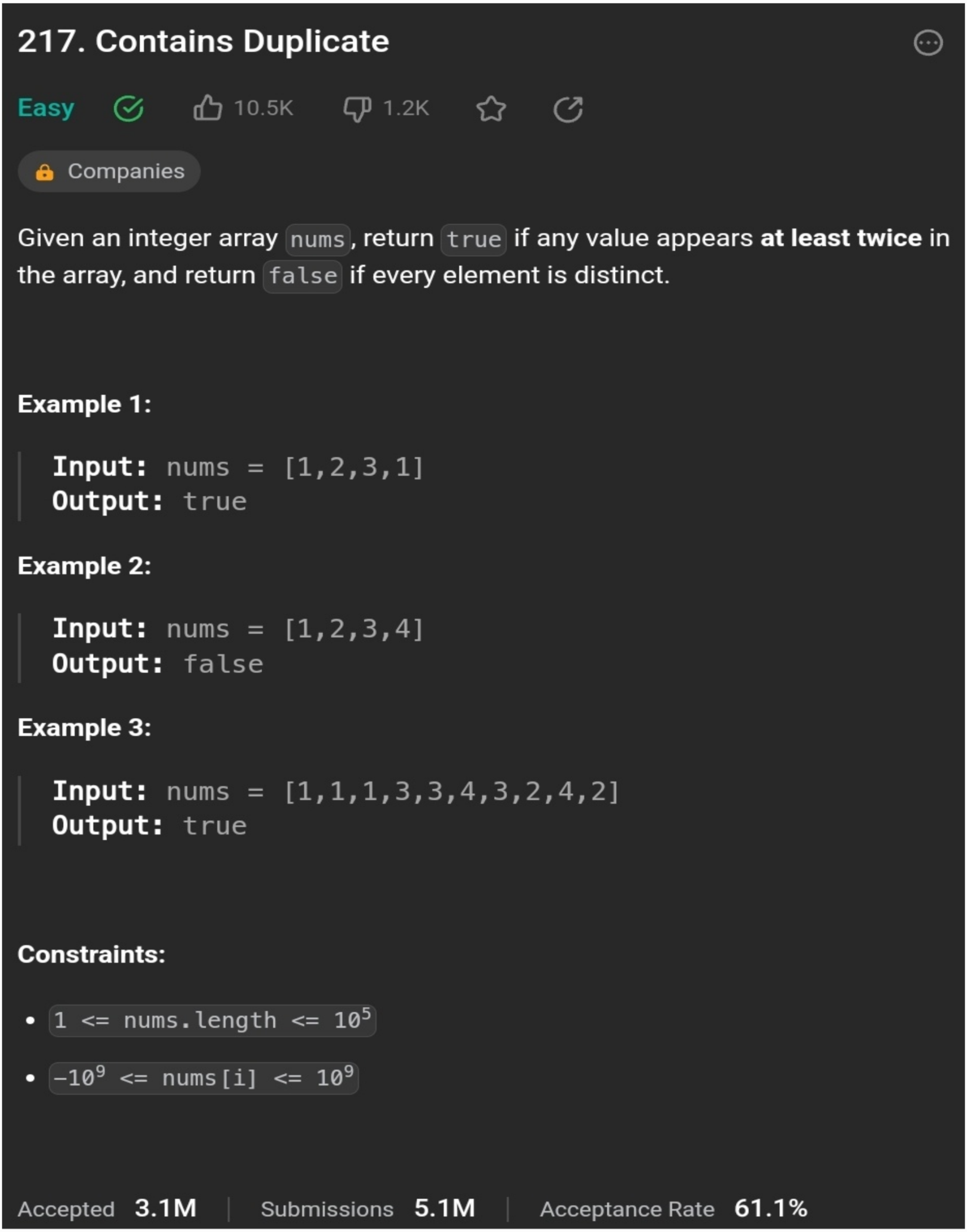

217. Contains Duplicate

Given an integer array `nums`, return `true` if any value appears **at least twice** in the array, and return `false` if every element is distinct.

Example 1:

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

Example 2:

```
Input: nums = [1,2,3,4]
Output: false
```

Example 3:

```
Input: nums = [1,1,1,3,3,4,3,2,4,2]
Output: true
```

Constraints:

- $1 <= \text{nums.length} <= 10^5$
- $-10^9 <= \text{nums[i]} <= 10^9$

Accepted 3.1M | Submissions 5.1M | Acceptance Rate 61.1%

Bruteforce:

$T(n) : O(n^2)$

$S(n) : O(1)$

Approach1:

Sort and then check.

$T(n) : O(n\log n)$
$S(n)$: Sorting Space

Approach 2:

Using hashset or hashmap

By using hashset we can write single line of code.

```
return (nums.size() > unordered_set<int>(
        nums.begin(), nums.end()).
                  size());
```

constructing a set from vector using range converter.
(or) we can also use copy() function.

(or)

we can traditionally construct a set from vector and see.

```
unordered_set<int> s;
for(auto i : nums)
    if(s.find(i) != s.end())
        return true
    else
        s.insert(i).
```

return false.