

LeetCode

Explore

Problems

Mock

Contest

Articles

Discuss

Store

Premium

Back to Chapter

Week 1: May 1st–May 7th

Problems appear at midnight, Pacific Time

First Bad Version

Jewels and Stones

Ransom Note

Number Complement

First Unique Character in a String

Majority Element

Cousins in Binary Tree

Week 2: May 8th–May 14th

Problems appear at midnight, Pacific Time

Week 3: May 15th–May 21st

The first problem for this section is locked

Previous

Next

Go to Discuss

Majority Element

Solution

Given an array of size n , find the majority element. The majority element is the element that appears **more than** $\lfloor n/2 \rfloor$ times.

You may assume that the array is non-empty and the majority element always exist in the array.

Example 1:

Input: [3,2,3]
Output: 3

Example 2:

Input: [2,2,1,1,1,2,2]
Output: 2

Java

```
1 class Solution {
2     public int majorityElement(int[] nums) {
3         Map<Integer,Integer> map=new HashMap<>();
4         for(int n:nums){
5             Integer x=map.get(n);
6             if(x==null) map.put(n,1);
7             else map.put(n,x+1);
8         }
9         for(Map.Entry it:map.entrySet()){
10             int max=(int)it.getValue();
11             if(max>=(float)nums.length/2){
12                 return (int)it.getKey();
13             }
14         }
15         return 0;
16     }
17 }
```

Custom Testcase (Contribute)

Run Code

Submit