169. Majority Element

Easy 8282 311 Add to List Share

Given an array `nums` of size `n`, return *the majority element*.

The majority element is the element that appears more than `⌊n / 2⌋` times. You may assume that the majority element always exists in the array.

Example 1:

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

Example 2:

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

Constraints:

- `n == nums.length`
- `1 <= n <= 5 * 10^4`
- `-2^31 <= nums[i] <= 2^31 - 1`

Follow-up: Could you solve the problem in linear time and in `O(1)` space?

Accepted 1,103,889 Submissions 1,774,612

Seen this question in a real interview before? Yes No

Companies

Related Topics

Java Autocomplete

```java
class Solution {

    public int majorityElement( int[] nums ) {

        int length = nums.length;
        int shouldAppear = length / 2;

        if(length == 1)
            return nums[0];

        HashMap<Integer, Integer> hashMap = new HashMap<>();

        for (int num : nums) {
            if (hashMap.containsKey(num)) {

                int occurence = hashMap.get(num) + 1;

                if(occurence > shouldAppear)
                    return num;

                hashMap.put(num, occurence);
            } else {
                hashMap.put(num, 1);
            }
        }

        return -1;
    }

}
```

Testcase Run Code Result Debugger

Accepted Runtime: 0 ms

Your input `[3,2,3]`

Output `3` Diff

Expected `3`