

# Problem 169: Majority Element

## Problem Information

**Difficulty:** Easy

**Acceptance Rate:** 65.96%

**Paid Only:** No

**Tags:** Array, Hash Table, Divide and Conquer, Sorting, Counting

## Problem Description

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

The majority element is the element that appears more than  $\lfloor n / 2 \rfloor$  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:**

`1 <= n <= 5 * 104` `-109 <= nums[i] <= 109` \* The input is generated such that a majority element will exist in the array.

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

## Code Snippets

**C++:**

```
class Solution {  
public:  
    int majorityElement(vector<int>& nums) {  
  
    }  
};
```

### Java:

```
class Solution {  
    public int majorityElement(int[] nums) {  
  
    }  
}
```

### Python3:

```
class Solution:  
    def majorityElement(self, nums: List[int]) -> int:
```