169. Majority Element

Easy | Topics | Companies

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`
- `-10^9 <= nums[i] <= 10^9`

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

Python:

```
from typing import List

class Solution:
    def majorityElement(self, nums: List[int]) -> int:
        count = 0
        candidate = None

        for num in nums:
            if count == 0:
                candidate = num
            count += (1 if num == candidate else -1)

        return candidate
```

JavaScript:

```
/**
 * @param {number[]} nums
 * @return {number}
 */
var majorityElement = function(nums) {
    let candidate = null;
    let count = 0;

    for (let num of nums) {
        if (count === 0) {
            candidate = num;  // choose a new candidate
        }
        count += (num === candidate) ? 1 : -1;
    }

    return candidate;
};
```

Java:

```
class Solution {
    public int majorityElement(int[] nums) {
        int candidate = 0;
        int count = 0;

        for (int num : nums) {
            if (count == 0) {
```

```
            candidate = num;
        }
        if (num == candidate) {
            count++;
        } else {
            count--;
        }
    }

    return candidate;
  }
}
```