Majority Element

Find the element that appears more than n/2 times in an array.

The **Majority Element** in an array is the element that appears **more than ⌊ n/2 ⌋ times**, where `n` is the size of the array.

This means it occurs **strictly more than half** of the total elements.

Example

Array:
`[2, 2, 1, 1, 2, 2, 2]`
Size = 7 → n/2 = 3.5 → need ≥ **4 times**

Here:

- `2` appears **5 times** → ✓ Majority element

So answer = **2**

Best Approach (Boyer–Moore Voting Algorithm)

This is the **most efficient method**.

- Keep a **candidate** and a **count**
- If count becomes 0 → change candidate
- Increase count if same element, decrease if different

Algorithm Steps

1. Initialize:
2. `candidate = None`

3. `count = 0`
4. Traverse array:
    - If count = 0 → set candidate = current element
    - If element = candidate → count++
    - Else → count--
5. Final candidate is the majority element.

Code (Java)

```
public class MajorityElement {
    public static int findMajority(int[] nums) {
        int candidate = 0, count = 0;

        for (int num : nums) {
            if (count == 0) {
                candidate = num;
            }
            count += (num == candidate) ? 1 : -1;
        }
        return candidate;
    }

    public static void main(String[] args) {
        int[] arr = {2, 2, 1, 1, 2, 2, 2};
        System.out.println("Majority Element: " + findMajority(arr));
    }
}
```