136. Single Number

Easy | Topics | Companies | Hint

Given a **non-empty** array of integers `nums`, every element appears *twice* except for one. Find that single one.

You must implement a solution with a linear runtime complexity and use only constant extra space.

Example 1:

> **Input:** nums = [2,2,1]
>
> **Output:** 1

Example 2:

> **Input:** nums = [4,1,2,1,2]
>
> **Output:** 4

Example 3:

> **Input:** nums = [1]
>
> **Output:** 1

Constraints:

- $1 <= nums.length <= 3 * 10^4$
- $-3 * 10^4 <= nums[i] <= 3 * 10^4$
- Each element in the array appears twice except for one element which appears only once.

Python:

```
from typing import List

class Solution:
    def singleNumber(self, nums: List[int]) -> int:
        result = 0
        for num in nums:
            result ^= num   # XOR all numbers
        return result
```

JavaScript:

```
/**
 * @param {number[]} nums
 * @return {number}
 */
var singleNumber = function(nums) {
    let result = 0;
    for (let num of nums) {
        result ^= num; // XOR each number
    }
    return result;
};
```

Java:

```
class Solution {
    public int singleNumber(int[] nums) {
        int result = 0;
        for (int num : nums) {
            result ^= num; // XOR cancels out duplicates
        }
        return result;
    }
}
```