Problem 3158: Find the XOR of Numbers Which Appear Twice

Problem Information

Difficulty: Easy
Acceptance Rate: 78.36%
Paid Only: No
Tags: Array, Hash Table, Bit Manipulation

Problem Description

You are given an array `nums`, where each number in the array appears **either** __ once __ or __ twice.

Return the bitwise __`XOR` of all the numbers that appear twice in the array, or 0 if no number appears twice.

Example 1:

Input: nums = [1,2,1,3]

Output: 1

Explanation:

The only number that appears twice in `nums` is 1.

Example 2:

Input: nums = [1,2,3]

Output: 0

Explanation:

No number appears twice in `nums`.

Example 3:

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

Output: 3

Explanation:

Numbers 1 and 2 appeared twice. `1 XOR 2 == 3`.

Constraints:

* `1 <= nums.length <= 50` * `1 <= nums[i] <= 50` * Each number in `nums` appears either once or twice.

Code Snippets

C++:

```
class Solution {
public:
int duplicateNumbersXOR(vector<int>& nums) {

}
};
```

Java:

```
class Solution {
public int duplicateNumbersXOR(int[] nums) {

}
}
```

Python3:

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