Chalkboard XOR Game

We are given non-negative integers nums[i] which are written on a chalkboard. Alice and Bob take turns erasing exactly one number from the chalkboard, with Alice starting first. If erasing a number causes the bitwise XOR of all the elements of the chalkboard to become 0, then that player loses. (Also, we'll say the bitwise XOR of one element is that element itself, and the bitwise XOR of no elements is 0.)

Also, if any player starts their turn with the bitwise XOR of all the elements of the chalkboard equal to 0, then that player wins.

Return True if and only if Alice wins the game, assuming both players play optimally.

```
Example:
Input: nums = [1, 1, 2]
Output: false
Explanation:
Alice has two choices: erase 1 or erase 2.
If she erases 1, the nums array becomes [1, 2]. The bitwise XOR of all the elements o
f the chalkboard is 1 XOR 2 = 3. Now Bob can remove any element he wants, because Ali
ce will be the one to erase the last element and she will lose.
If Alice erases 2 first, now nums becomes [1, 1]. The bitwise XOR of all the elements
 of the chalkboard is 1 XOR 1 = 0. Alice will lose.
```

Notes:

- `0 <= N <= 1000`.
- `0 <= nums[i] <= 2^16`.

Solution 1

Why [1,2,3] returns true?
How can Alice win? No matter what she erase, Bob can erase any number and Alice will loss.

written by paulwater original link here

Solution 2

Should not be Hard problem

The solution can be only 3 lines and really simple.
It should be a medium problem.
`Hard` is kind of misleading.

Why `[1, 2, 3]` expect `true`

If `xor == 0` at first for Alice, it is considered as Bob lose already!
I find this stupid idea just after contest and I think it has no reason.
It should complete this condition in the description!
Update: @awice confirmed that "It was supposed to be added to the problem description that having an initial XOR of 0 is a win."

Let's discuss it if we add this condition.

If `xor == 0`, Alice win directly.
If `xor != 0` and length of numbers is even, Alice will win.

Beacause:
All numbers won't be the same. Otherwise `xor` will be equal to `0`
If all numbers are not the same, It means there are at least 2 different numbers.
Alice can always erase a number different from current `xor`.
So Alice won't never lose this turn at this situation.

If we don't have the condition

Just return `nums are not all 0` and `length of nums is even`

C++:

```
    bool xorGame(vector<int>& nums) {
        int xo = 0;
        for (int i: nums) xo ^= i;
        return xo == 0 || nums.size() % 2 == 0;
    }
```

Java:

```
    public boolean xorGame(int[] nums) {
        int xor = 0;
        for (int i: nums) xor ^= i;
        return xor == 0 || nums.length % 2 == 0;
    }
```

Python

```
    def xorGame(self, nums):
        xor = 0
        for i in nums: xor ^= i
        return xor == 0 or len(nums) % 2 == 0
```

written by lee215 original link here

Solution 3

The problem is meaningless. And the solution is tricky...

written by wangzi6147 original link here