# 题目

给你两个下标从 0 开始的数组 nums1 和 nums2 ，两个数组都只包含非负整数。请你求出另外一个数组 nums3 ，包含 nums1 和 nums2 中 所有数对 的异或和（nums1 中每个整数都跟 nums2 中每个整数 恰好 匹配一次）。

请你返回 nums3 中所有整数的 异或和 。

示例 1：

输入：nums1 = [2,1,3], nums2 = [10,2,5,0]

输出：13

解释：

一个可能的 nums3 数组是 [8,0,7,2,11,3,4,1,9,1,6,3] 。

所有这些数字的异或和是 13 ，所以我们返回 13 。

示例 2：

输入：nums1 = [1,2], nums2 = [3,4]

输出：0

解释：

所有数对异或和的结果分别为 nums1[0] ^ nums2[0] ，nums1[0] ^ nums2[1] ，nums1[1] ^ nums2[0] 和 nums1[1] ^ nums2[1] 。

所以，一个可能的 nums3 数组是 [2,5,1,6] 。

2 ^ 5 ^ 1 ^ 6 = 0 ，所以我们返回 0 。

提示：

1 <= nums1.length, nums2.length <= 105

0 <= nums1[i], nums2[j] <= 109

# 分析

只需知道一个性质和两个公式

性质：异或满足交换律

公式一：a^a=0，也就是说任何一个数和它本身异或的结果是0

公式二：a^0=a，也即使是任何一个数和0异或的结果是它本身

代码：

class Solution {

public:

int xorAllNums(vector<int>& nums1, vector<int>& nums2) {

int ans = 0;

if (nums1.size() % 2) {

for (auto i : nums2) {

ans ^= i;

}

}

if (nums2.size() % 2) {

for (auto i : nums1) {

ans ^= i;

}

}

return ans;

}

};