# 描述

给定一个非空整数数组，除了某个元素只出现一次以外，其余每个元素均出现两次。找出那个只出现了一次的元素。

**说明：**

你的算法应该具有线性时间复杂度。 你可以不使用额外空间来实现吗？

**示例 1:**

输入: [2,2,1]

输出: 1

**示例 2:**

输入: [4,1,2,1,2]

输出: 4

# 分析

方法 1：列表操作

遍历 \text{nums}nums 中的每一个元素

如果某个 \text{nums}nums 中的数字是新出现的，则将它添加到列表中

如果某个数字已经在列表中，删除它

方法 2：哈希表

我们用哈希表避免每次查找元素是否存在需要的 O(n)O(n) 时间。

遍历 \text{nums}nums 中的每一个元素

查找 hash\\_tablehash\_table 中是否有当前元素的键

如果没有，将当前元素作为键插入 hash\\_tablehash\_table

最后， hash\\_tablehash\_table 中仅有一个元素，用 popitem 获得它

方法 3位操作

如果我们对 0 和二进制位做 XOR 运算，得到的仍然是这个二进制位

a⊕0 =a

如果我们对相同的二进制位做 XOR 运算，返回的结果是 0

a⊕a=0

XOR 满足交换律和结合律

a⊕b⊕a=(a⊕a)⊕b=0⊕b=b

所以我们只需要将所有的数进行 XOR 操作，得到那个唯一的数字。

# 代码