# 56. 数组中只出现一次的数字

## 题目链接

[牛客网](https://www.nowcoder.com/practice/389fc1c3d3be4479a154f63f495abff8?tpId=13&tqId=11193&tab=answerKey&from=cyc_github)

## 题目描述

一个整型数组里除了两个数字之外，其他的数字都出现了两次，找出这两个数。

## 解题思路

两个相等的元素异或的结果为 0，而 0 与任意数 x 异或的结果都为 x。

对本题给的数组的所有元素执行异或操作，得到的是两个不存在重复的元素异或的结果。例如对于数组 [x,x,y,y,z,k]，xxyyz^k = 0yyzk = yyz^k = 0zk = z^k。

两个不相等的元素在位级表示上一定会有所不同，因此这两个元素异或得到的结果 diff 一定不为 0。位运算 diff & -diff 能得到 diff 位级表示中最右侧为 1 的位，令 diff = diff & -diff。将 diff 作为区分两个元素的依据，一定有一个元素对 diff 进行异或的结果为 0，另一个结果非 0。设不相等的两个元素分别为 z 和 k，遍历数组所有元素，判断元素与 diff 的异或结果是否为 0，如果是的话将元素与 z 进行异或并赋值给 z，否则与 k 进行异或并赋值给 k。数组中相等的元素一定会同时与 z 或者与 k 进行异或操作，而不是一个与 z 进行异或，一个与 k 进行异或。而且这些相等的元素异或的结果为 0，因此最后 z 和 k 只是不相等的两个元素与 0 异或的结果，也就是不相等两个元素本身。

下面的解法中，num1 和 num2 数组的第一个元素是用来保持返回值的... 实际开发中不推荐这种返回值的方式。

public int[] FindNumsAppearOnce (int[] nums) {
 int[] res = new int[2];
 int diff = 0;
 for (int num : nums)
 diff ^= num;
 diff &= -diff;
 for (int num : nums) {
 if ((num & diff) == 0)
 res[0] ^= num;
 else
 res[1] ^= num;
 }
 if (res[0] > res[1]) {
 swap(res);
 }
 return res;
}

private void swap(int[] nums) {
 int t = nums[0];
 nums[0] = nums[1];
 nums[1] = t;
}