



链滴

数组中只出现一次的数字

作者: [yudake](#)

原文链接: <https://ld246.com/article/1518921395610>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

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。请写程序找出这两个只出现一次的数字。

num1,num2分别为长度为1的数组。传出参数
将num1[0],num2[0]设置为返回结果

解题思路

两种方法：

- 只能遍历一次
 - 维护一个list或者map，记录本数字是否只出现一次。
- 不能创建新空间，或者空间复杂度为 $O(1)$ 。
 - 数组中所有数字异或，最后的结果是两个数字的异或。
 - 然后把结果数字转换成二进制，二进制中找到一个1的位置N，说明两个数字一个在位置N是1，一个在位置N是0
 - 然后对数组中位置N是1的所有数字进行异或，得到数字1
 - 再对数组中位置N是0的所有数字异或，得到数字2

空间复杂度 $O(1)$ ，时间复杂度 $O(2n)$ 。

代码

```
public class Solution {
    public void FindNumsAppearOnce(int [] array,int num1[] , int num2[]) {
        int tmp = 0;
        num1[0] = 0;
        num2[0] = 0;
        for (int i = 0; i < array.length; i++)
            tmp ^= array[i];
        int index = findFirstBitIs(tmp);
        for (int i = 0; i < array.length; i++)
            if (isBit(array[i], index))
                num1[0] ^= array[i];
            else
                num2[0] ^= array[i];
    }

    public int findFirstBitIs(int num){
        int indexBit = 0;
        while(((num & 1)==0) && (indexBit)<8*4){
            num = num >> 1;
            ++indexBit;
        }
        return indexBit;
    }
}
```

```
public boolean isBit(int num,int indexBit){  
    num = num >> indexBit;  
    return (num & 1) == 1;  
}  
}
```