

# 剑指 Offer 56 - II. 数组中数字出现的次数 II

Wednesday, June 15, 2022 10:00 AM

<https://leetcode.cn/problems/shu-zu-zhong-shu-zi-chu-xian-de-ci-shu-ii-lcof/>

中等

在一个数组 `nums` 中除一个数字只出现一次之外，其他数字都出现了三次。请找出那个只出现一次的数字。

示例 2:

输入: `nums = [9,1,7,9,7,9,7]`

输出: 1

思路:

1. 将每个数的二进制每一位都加起来。结果存至长度为32的数组 `res[]` 中。
2. 这样, `res[i]%3==0` 意味着:  
其他重复三位的数, 此位为0或1, 累加后的`res[i]`均能整除3;  
故目标数 `x` 的此位为0。  
`res[i]%3==1` 意味着:  
其他重复三位的数, 此位为0或1, 累加后的`res[i]`均能整除3;  
此时余1, 故目标数 `x` 的此位为1。

ps: 据此模式, 可以解其他数字出现任意次而目标数只出现一次的情况。

code:

```
class Solution {
    public int singleNumber(int[] nums) {
        // 整数为4字节
        int[] k=new int[32];
        // 累加每个数的每一位
        for(int i=0;i<nums.length;i++){
            // 每个数 与1相与 结果取决于最后一位是1还是0
            // 因为1的二进制为000000...001
            for(int j=0;j<32;j++){
                k[j] += (nums[i]>>j &1)==1?1:0;
            }
        }
        // res初始化为0
        int res=0;
        // 左移, 故倒着处理 res[]
        for(int i=31;i>=0;i--){
            res=res<<1;
            if(k[i]%3==1){
                // 此时目标数该位为1, 故与该位 与1相或,
                // 不影响已经赋值的其他位且把此位赋为1
            }
        }
    }
}
```

```

        res=(res|1);
    }
    // 该位为0不处理，直接左移即可。
}
return res;
}
}

```

介于此题与上题类似，便于理解，增加简单题一道，超级简单。

## 剑指 Offer 57. 和为s的两个数字

<https://leetcode.cn/problems/he-wei-sde-liang-ge-shu-zi-lcof/>

简单

输入一个递增排序的数组和一个数字s，

在数组中查找两个数，

使得它们的和正好是s。

如果有多对数字的和等于s，则输出任意一对即可。

示例 1：

输入：nums = [2,7,11,15], target = 9

输出：[2,7] 或者 [7,2]

我的思路：

双指针，左右遍历，因为肯定存在、数组递增

故大了移动右指针，小了移左指针即可。

code：

```

class Solution {
    public int[] twoSum(int[] nums, int target) {
        int l=0;
        int r=nums.length-1;

        while(l<r){
            if(nums[l]+nums[r]==target){
                return new int[]{nums[l],nums[r]};
            }else if(nums[l]+nums[r]>target){
                r--;
            }else{
                l++;
            }
        }
        return new int[]{};
    }
}

```

