

### 数组中数字出现的次数

一个整型数组 `nums` 里除两个数字之外，其他数字都出现了两次。请写程序找出这两个只出现一次的数字。要求时间复杂度是  $O(n)$ ，空间复杂度是  $O(1)$ 。

示例 1:

输入: `nums = [4,1,4,6]`

输出: `[1,6]` 或 `[6,1]`

示例 2:

输入: `nums = [1,2,10,4,1,4,3,3]`

输出: `[2,10]` 或 `[10,2]`

1. 第一次全部异或

0010	② ✓
0100	④ ✗
0110	
0011	③ ✓
0101	
0110	⑥ ✓
0011	
0011	③ ✓
0000	
0010	② ✓
0010	
0101	⑤ ✗
0111	
0101	⑤ ✗
0010	

2.1 分组第一组异或

0010	②
0011	③
0001	
0110	⑥
0111	
0011	③
0100	
0010	②
0110	⇒ 6

2.2 第二组异或同理  
得出 5

按照二进制数第 2 位为 1 进行分组:  
所以 ② ③ ⑥ ③ ② 一组  
④ ⑤ ⑤ 一组  
然后依次异或, 转上面 2.1, 2.2, 即得到两个只出现一次的数

C:

```
int* singleNumbers(int* nums, int numsSize, int* returnSize){
    int res=0;
    for(int i=0;i<numsSize;i++)
    {
        res^=nums[i];
    }
    res=res&(-1*res);
    int a=0,b=0;
    for(int i=0;i<numsSize;i++)
    {
        if((nums[i] & res)==0)
        {
            a^=nums[i];
        }
        else
        {
            b^=nums[i];
        }
    }
    *returnSize=2;
    int* ret=(int*)malloc(2*sizeof(int)); //根据返回值类型 创建存储
    ret[0]=a;
    ret[1]=b;
    return ret;
}
```

C++:

```
class Solution {
public:
    vector<int> singleNumbers(vector<int>& nums) {
        vector<int> ret;
        int res=0;
        for(int i=0;i<nums.size();i++)
        {
            res^=nums[i];
        }
        res=res&(-1*res);
        int a=0,b=0;
        for(int i=0;i<nums.size();i++)
        {
            if((nums[i]&res)==0)
            {
                a^=nums[i];
            }
        }
    }
};
```

```
        }  
        else  
        {  
            b^=nums[i];  
        }  
    }  
    return vector<int>{a,b};  
}  
};
```