

题目一 数组中只出现一次的两个数字 要求时间 $O(n)$ ，空间 $O(1)$

面试题 56：数组中数字出现的次数

题目一：数组中只出现一次的两个数字。

一个整型数组里除两个数字之外，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 $O(1)$ 。

解 位运算 异或

首先把数组所有的数字异或，结果肯定就是只出现一次的两个数字的异或结果。这个结果肯定不为0，我们找到这个数的右边第一个1，以该位是否为1将数组分成两个部分，那么就变成了 分别含只出现一次的一个数字 的两个数组

```
void findNums(vector<int> data,int *num1,int *num2)
{
    if(data.empty())
        return;
    int num=data[0];
    for(int i=1;i<data.size();i++)
        num^=data[i];
    int index=findFirst1(num);
    *num1=*num2=0;
    for(int j=0;j<data.size();j++)
    {
        if(isBit1(data[j],index))
            *num1^=data[j];
        else
            *num2^=data[j];
    }
}

int findFirst1(int num)
{
    int index=0;
    while(((num&1)==0) && (index<8*sizeof(int))) //num&1 一定要有括号
    {
        num=num>>1;
        index++;
    }
    return index;
}

bool isBit1(int num,int index)
{
    }
```

```
num>>=index;
return (num&1);
}
```

题目二 数组中唯一只出现一次的数字

题目二：数组中唯一只出现一次的数字。

在一个数组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。

解

用一个32位数组记录每一位出现的次数之和，时间 $O(n)$ ，空间 $O(1)$ 。哪一位的次数和不能整除3，说明只出现一次的那个数在该位上为1

```
int findNumber(vector<int> nums)
```

```
{
    if(nums.empty())
    {
        return -1;
    }
    vector<int> bitSum(32,0);
    for(int i=0;i<nums.size();i++)
    {
        int bitMask=1;
        for(int j=31;j>=0;j--)
        {
            int bit=nums[i]&bitMask;
            if(bit!=0)
                bitSum[j]+=1;
            bitMask<<=1;
        }
    }
    int res=0;
    for(int i=0;i<32;i++)
    {
        res=res<<1;
        res+=(bitSum[i]%3);
    }
    return res;
}
```