

## 错误的集合

集合  $s$  包含从 1 到  $n$  的整数。不幸的是，因为数据错误，导致集合里面某一个数字复制了成了集合里面的另外一个数字的值，导致集合 **丢失了一个数字** 并且 **有一个数字重复**。

给定一个数组 `nums` 代表了集合  $s$  发生错误后的结果。

请你找出重复出现的整数，再找到丢失的整数，将它们以数组的形式返回。

示例 1:

输入: `nums = [1,2,2,4]`

输出: `[2,3]`

示例 2:

输入: `nums = [1,1]`

输出: `[1,2]`

```
class Solution {
public:
    vector<int> findErrorNums(vector<int>& nums) {
        sort(nums.begin(), nums.end()); // 先排序, 保证重复元素相邻
        int len=nums.size();
        vector<int> ans;
        int repe=-1; // 记录重复元素
        int lost=-1; // 记录丢失元素
        for(int j=0; j<len-1; j++)
        {
            if(nums[j]==nums[j+1])
            {
                repe=nums[j];
            }
            if(nums[j+1]-nums[j]>1)
            {
                lost=nums[j]+1;
            }
            if(repe!=-1&&lost!=-1)
                break;
        }
        // 先放重复元素, 再放丢失元素 【1, 3, 3】 -> 【3, 2】
        if(repe!=-1)
            ans.push_back(repe);
        if(lost!=-1)
```

```
        ans.push_back(lost);  
        //特殊情况:  
        //对于【1,1】，【2,2】输入,上述操作只能找到重复元素，则缺失元素处理如下:  
        //缺失逻辑根据编译结果得到  
        if(ans.size()<2)  
        {  
            if(nums[0]!=1)  
            {  
                ans.push_back(1);//【2,2】->【2,1】  
            }  
            else  
            {  
                ans.push_back(nums.size());//【1,1】->【1,2】  
            }  
        }  
        return ans;  
    }  
};
```