

给定一个范围在  $1 \leq a[i] \leq n$  ( $n$  = 数组大小) 的 整型数组，数组中的元素一些出现了两次，另一些只出现一次。

找到所有在  $[1, n]$  范围之间没有出现在数组中的数字。

您能在不使用额外空间且时间复杂度为  $O(n)$  的情况下完成这个任务吗？你可以假定返回的数组不算在额外空间内。

**示例:**

**输入:**

[4,3,2,7,8,2,3,1]

**输出:**

[5,6]

思路，按题目理解，有的出现了两次，有的出现1次，用出现的数字，把其对应的位置的数变为负数，因为有出现两次的，所以用绝对值处理，然后再遍历数组，当前位置为正的，那么就未在数组中出现过，添加到返回的数组即可。

```
class Solution {
public:
    vector<int> findDisappearedNumbers(vector<int>& nums) {
        vector<int> result;
        int n=nums.size();
        for(int i=0;i<n;i++){
            int n=abs(nums[i]);
            nums[n-1]=-1*abs(nums[n-1]);
        }
        for(int i=0;i<n;i++){
            if(nums[i]>0){
                result.push_back(i+1);
            }
        }
        return result;
    }
};
```