

## 442 数组中重复的数据

Label: 数组

给定一个整数数组  $a$ ，其中  $1 \leq a[i] \leq n$ （ $n$ 为数组长度），其中有些元素出现两次而其他元素出现一次。

找到所有出现两次的元素。

你可以不用到任何额外空间并在 $O(n)$ 时间复杂度内解决这个问题吗？

- API 排序 查找

```
class Solution {
    public List<Integer> findDuplicates(int[] nums) {
        Arrays.sort(nums);
        List<Integer> re = new ArrayList<>();
        for (int i = 0; i < nums.length - 1; i++) {
            if (nums[i] == nums[i+1]) {
                re.add(nums[i]);
                i++; // 只会出现两次
            }
        }
        return re;
    }
}
```

- 使用标记位

// 数组中具有相同值的元素，会被经过两次。

// 那么只要想出一种方式在这个遍历结束后可以区分，哪些元素被经过了多次即可

```
class Solution {
    public List<Integer> findDuplicates(int[] nums) {
        List<Integer> re = new ArrayList<>();
        for (int i = 0; i < nums.length; i++) {
            int num = Math.abs(nums[i]); // 怕该值被之前的index反转标记过
            if (nums[num - 1] > 0) {
                nums[num - 1] *= -1; // 标记为 负数
            } else {
                re.add(num); // 如果下次又值再寻址到这个位置,说明这个index重复
            }
        }
        return re;
    }
}
```

- 类似使用标记

```
class Solution {
    public List<Integer> findDuplicates(int[] nums) {
        int n = nums.length;
        List<Integer> ans = new ArrayList<>();
        for (int i : nums) {
            nums[(i - 1) % n] = nums[(i - 1) % n] + n; // 出现一次就在响应地方加 n
        }

        for (int i = 0; i < n; i++) {
            if(nums[i] < 3*n + 1 && nums[i] > 2*n) {
                ans.add(i+1);
            }
        }
        return ans;
    }
}
```