

169 多数元素

Label: 数组、位运算

给定一个大小为 n 的数组，找到其中的多数元素。多数元素是指在数组中出现次数 大于 $\lfloor n/2 \rfloor$ 的元素。
你可以假设数组是非空的，并且给定的数组总是存在多数元素。

输入: [3,2,3]

输出: 3

输入: [2,2,1,1,1,2,2]

输出: 2

- 排序

```
class Solution {
    public int majorityElement(int[] nums) {
        Arrays.sort(nums);
        return nums[nums.length/2];
    }
}
```

- 统计查找

```
class Solution {
    public int majorityElement(int[] nums) {
        Map<Integer,Integer> map = new HashMap<>();
        // 统计
        for (int i : nums) {
            map.put(i, map.getOrDefault(i, 0)+1);
        }
        // 查找
        for (int i : map.keySet()) {
            if (map.get(i) > nums.length/2) {
                return i;
            }
        }
        return -1;
    }
}
```

- 摩尔投票法（这就相当于每个“多数元素”和其他元素 两两相互抵消，抵消到最后肯定还剩余至少 1 个“多数元素”）

```
class Solution {
    public int majorityElement(int[] nums) {
        int cand_num = nums[0], count = 1;

        for (int i = 1; i < nums.length; ++i) {
            if (cand_num == nums[i]) // 遇到相同数，票数 +1
                ++count;

            else if (--count == 0) { // 重置候选人
                cand_num = nums[i];
                count = 1;
            }
        }
        return cand_num;
    }
}
```

- 位运算（没太看懂）

```
class Solution {
    // 由于众数数组中出现次数大于n/2，那么众数对应的二进制每一个为1的位数出现的次数一定大于n/2
    public int majorityElement(int[] nums) {
        int result = 0, k = nums.length >> 1;
        for (int j = 0; j < 32; j++) {
            int count = 0;
            for (int num : nums) {
                count += num >> j & 1;
                if (count > k) {
                    result += 1 << j;
                    break;
                }
            }
        }
        return result;
    }
}
```