**难度: 简单**

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 ⌊ n/2 ⌋ 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。

示例 1:

输入: [3,2,3]

输出: 3

示例 2:

输入: [2,2,1,1,1,2,2]

输出: 2

来源：力扣（LeetCode）

链接：https://leetcode-cn.com/problems/majority-element

著作权归领扣网络所有。商业转载请联系官方授权，非商业转载请注明出处。

**思路1(使用hashmap):**

1. 用hashmap通过key获取对应的value是否为空

2. 为空就通过通过put,key赋值1

3. 不为空就value+1并且判断一下是否大于等于数组长度的一半

4.一半直接返回对应的数组,不一半就进行put

|  |
| --- |
| public int majorityElement(int[] nums) {  HashMap<Integer,Integer> map = new HashMap<>();  for (int i = 0; i < nums.length; i++) {  Integer value = map.get(nums[i]);  value = value == null ? 0 : value + 1;  if (value >= nums.length / 2) {  return nums[i];  }else {  map.put(nums[i],value);  }  }  return -1;  } |

思路2(使用摩尔算法):

1). 先创建一个计数器为1

2). 每次如果是相同的计数器就+1

3). 不同计数器就-1

4). 如果计数器为0就替换为1,并且转换结果

|  |
| --- |
| public int majorityElement2(int[] nums) {  int count = 1;  int res = nums[0];  for (int i = 1; i < nums.length; i++) {  if (nums[i] == res) {  count++;  }else {  count--;  }  if (count < 1) {  count = 1;  res = nums[i];  }  }  return res;  } |