**难度: 简单**

数组中有一个数字出现的次数超过数组长度的一半，请找出这个数字。

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: [1, 2, 3, 2, 2, 2, 5, 4, 2]

输出: 2

限制：

1 <= 数组长度 <= 50000

来源：力扣（LeetCode）

链接：https://leetcode-cn.com/problems/shu-zu-zhong-chu-xian-ci-shu-chao-guo-yi-ban-de-shu-zi-lcof

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(使用摩尔投票算法):**

1). 判断两个值是否相同,相同就总数加1,不相同就总数减1

2). 当总数等于0的时候,就把存在多数的元素换成另一个,然后就把总数变成1

3). 最后直接把元素返回

**注:** 因为每次都会抵消,所有就会留下哪个最多的哪个

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