# 39. 数组中出现次数超过一半的数字

[NowCoder](https://www.nowcoder.com/practice/e8a1b01a2df14cb2b228b30ee6a92163?tpId=13&tqId=11181&tPage=1&rp=1&ru=/ta/coding-interviews&qru=/ta/coding-interviews/question-ranking&from=cyc_github)

## 解题思路

多数投票问题，可以利用 Boyer-Moore Majority Vote Algorithm 来解决这个问题，使得时间复杂度为 O(N)。

使用 cnt 来统计一个元素出现的次数，当遍历到的元素和统计元素相等时，令 cnt++，否则令 cnt--。如果前面查找了 i 个元素，且 cnt == 0，说明前 i 个元素没有 majority，或者有 majority，但是出现的次数少于 i / 2 ，因为如果多于 i / 2 的话 cnt 就一定不会为 0 。此时剩下的 n - i 个元素中，majority 的数目依然多于 (n - i) / 2，因此继续查找就能找出 majority。

public int MoreThanHalfNum\_Solution(int[] nums) {
 int majority = nums[0];
 for (int i = 1, cnt = 1; i < nums.length; i++) {
 cnt = nums[i] == majority ? cnt + 1 : cnt - 1;
 if (cnt == 0) {
 majority = nums[i];
 cnt = 1;
 }
 }
 int cnt = 0;
 for (int val : nums)
 if (val == majority)
 cnt++;
 return cnt > nums.length / 2 ? majority : 0;
}