/\*

Given an array of size n, find the majority element. The majority element is the element that appears more than ⌊ n/2 ⌋ times.

You may assume that the array is non-empty and the majority element always exist in the array.

way-1:用一个map来存出现的次数

way-2：Moore voting algorithm--每找出两个不同的element，就成对删除即count--，最终剩下的一定就是所求的。

\*/

class Solution {

public:

int majorityElement(vector<int>& nums)

{

//way-1

/\*

map<int,int> count;

for(int i=0;i<nums.size();i++)

{

if((++count[nums[i]])>nums.size()/2)

return nums[i];

}

return 0;

\*/

//way-2

int candidate = 0;

int count = 0;

for(int i = 0; i < nums.size(); i ++)

{

if(count == 0)

{

candidate = nums[i];

count++;

}

else

{

if(nums[i] == candidate)

count ++;

else

count --;

}

}

return candidate;

}

};