[079]169. Majority Element

- Date:2020-12-07(Mon)
- easy
- Related topic:array, divide and conquer, bit manipulation

problem link:https://leetcode.com/problems/majority-element/

> 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.

Example 1:

- Input: [3,2,3]
- Output: 3

Example 2:

- Input: [2,2,1,1,1,2,2]
- Output: 2

Think Process:

- 本題是經典的 **“博耶-摩爾多數投票算法（英語：Boyer–Moore majority vote algorithm)"**，也就是說把整個nums分為兩個part
    - 1majority
    - 2non-majority

並且，遇到count==0時，就把num[i]的值設為candidate，即使會變動，但是只要是占多數，最後則會輸出該candidate

code:

```
class Solution {
    public int majorityElement(int[] nums) {
        int candidate = 0;
        int count =0;
        for(int i=0;i<nums.length;i++){
            if(count==0){
                candidate = nums[i];
            }
            if(nums[i]==candidate) count++;
            else{
                count--;
            }
        }
        return candidate;
    }
}
```

Complexity Analysis

- Time complexity : O(n)

    - Boyer-Moore performs constant work exactly nn times, so the algorithm runs in linear time.

- Space complexity : O(1)

    - Boyer-Moore allocates only constant additional memory.