

```

import Foundation
/// 剑指 Offer 39. 数组中出现次数超过一半的数字
/// 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。
class Solution {
    /// 执行用时：84 ms, 在所有 Swift 提交中击败了 68.75% 的用户
    /// 内存消耗：15.5 MB, 在所有 Swift 提交中击败了 78.13% 的用户
    /// 通过测试用例：45 / 45
    func majorityElement(_ nums: [Int]) -> Int {
        /// 先随便选一个数认为是出现最多的数
        var result = nums[0]
        /// 设置出现次数为1
        var counter = 1
        /// 遍历的时候要去除掉第一个元素
        for num in nums.dropFirst() {
            if num == result {
                /// 如果遇到相同的数出现次数`+1`
                counter += 1
            } else {
                /// 如果遇到不相同的数出现次数`-1`
                counter -= 1
                /// 如果出现次数变为0了则换成当前的数字
                /// 并重新开始计数
                if (counter == 0) {
                    result = num
                    counter += 1
                }
            }
        }
        /// 无论如何都能决出胜负
        return result
    }
}

let solution = Solution()
print(solution.majorityElement([1, 2, 9, 2, 4, 4, 4, 4, 5]))

```