

169. 求众数

题目: Majority Element

语言: python3

英文版链接: <https://leetcode.com/problems/majority-element/description/>

中文版链接: <https://leetcode-cn.com/problems/majority-element/>

题目分析

求众数是一个经典的数学问题, 先对数组排序, 最中间那个数出现次数一定多于 $n / 2$ 。

答案

```
def majorityElement(self, nums) -> int:
    nums = sorted(nums)
    return nums[len(nums) // 2]
```

答案2

本题还可以利用 Boyer-Moore Majority Vote Algorithm 来解决这个问题, 使得时间复杂度为 $O(N)$ 。可以这么理解该算法: 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不相等时, 令 $\text{cnt} -= 1$ 。如果前面查找了 i 个元素, 且 $\text{cnt} == 0$, 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。

```
def majorityElement2(self, nums) -> int:
    cnt = 0
    majority = nums[0]
    for num in nums:
        if cnt == 0:
            majority = num
        if majority == num:
            cnt += 1
        else:
            cnt -= 1
    return majority
```