229. Majority Element II

Given an integer array of size `n`, find all elements that appear more than `⌊ n/3 ⌋` times.

Example 1:

```
Input: nums = [3,2,3]
Output: [3]
```

Example 2:

```
Input: nums = [1]
Output: [1]
```

Example 3:

```
Input: nums = [1,2]
Output: [1,2]
```

Constraints:

- `1 <= nums.length <= 5 * 10<sup>4</sup>`
- `-10<sup>9</sup> <= nums[i] <= 10<sup>9</sup>`

Follow up: Could you solve the problem in linear time and in `O(1)` space?

```python
class Solution:
    def majorityElement(self, nums: List[int]) -> List[int]:
        val1=nums[0]
        count1 = 1
        val2 = nums[0]
        count2 = 0

        for i in range(1,len(nums)):
            if nums[i]==val1:
                count1+=1
            elif nums[i]==val2:
                count2+=1
            else:
                if count1==0:
                    val1=nums[i]
                    count1+=1
                elif count2==0:
```

```
                val2 = nums[i]
                count2+=1
            else:
                count1-=1
                count2-=1
        ans = []
        c1 = 0
        c2 = 0
        for ele in nums:
            if ele==val1:
                c1+=1
            elif ele==val2:
                c2+=1
        if c1>len(nums)//3:
            ans.append(val1)
        if c2>len(nums)//3:
            ans.append(val2)
        return ans
```