Majority Vote (Ver 2)

solved by	Senan
Platform	GeeksForGeeks
difficulty	Medium
tags	Moore Algorithm
language	C++
solved on	@20/11/2024
link	https://www.geeksforgeeks.org/problems/majority-vote/1
Completion	☑

Intuition

To find the majority elements that appear more than `n/3` times in an array of integers, we can use the **Boyer-Moore Voting Algorithm** optimized for the scenario where the majority count is greater than `n/3`. The algorithm maintains two potential candidates and their respective counts, as there can be at most two such elements.

Approach

1. Traverse the array while maintaining two candidates and their counts.
2. In the first pass:
    - If the current number matches one of the candidates, increment its count.
    - If there’s an available candidate slot (count = 0), assign the current number to that slot.
    - If neither slot is available and no match, decrement both counts.
3. In the second pass:
    - Verify whether these candidates appear more than `n/3` times by counting their occurrences.

Complexity

Time Complexity:

The algorithm makes two passes through the array:

1. First pass: Identify the two potential majority candidates (O(n)).
2. Second pass: Verify their counts (O(n)).

Thus, the overall time complexity is: $O(n)$

Space Complexity:

The algorithm uses a constant amount of extra space for storing candidates and their counts. Hence, the space complexity is: $O(1)$

Code

```
class Solution {
  public:
    vector<int> findMajority(vector<int>& nums) {
        int cand1 = INT_MIN, cand2 = INT_MIN;
        int cnt1 = 0, cnt2 = 0;
        int minVotes = nums.size()/3;

        for(auto num: nums){
            if(cand1 == num) cnt1++;
            else if(cand2 == num) cnt2++;
            else if(cnt1 == 0){
                cand1 = num;
                cnt1 = 1;
            }
            else if(cnt2 == 0){
                cand2 = num;
                cnt2 = 1;
            }
            else{
                cnt1--;
                cnt2--;
            }
        }
        vector<int> answer;
        if(count(nums.begin(),nums.end(),cand1) > minVotes) answer.push_back(cand1);
        if(count(nums.begin(),nums.end(),cand2) > minVotes) answer.push_back(cand2);

        if(answer.size()==0) return {};
        sort(answer.begin(), answer.end());
        return answer;
    }
};
```