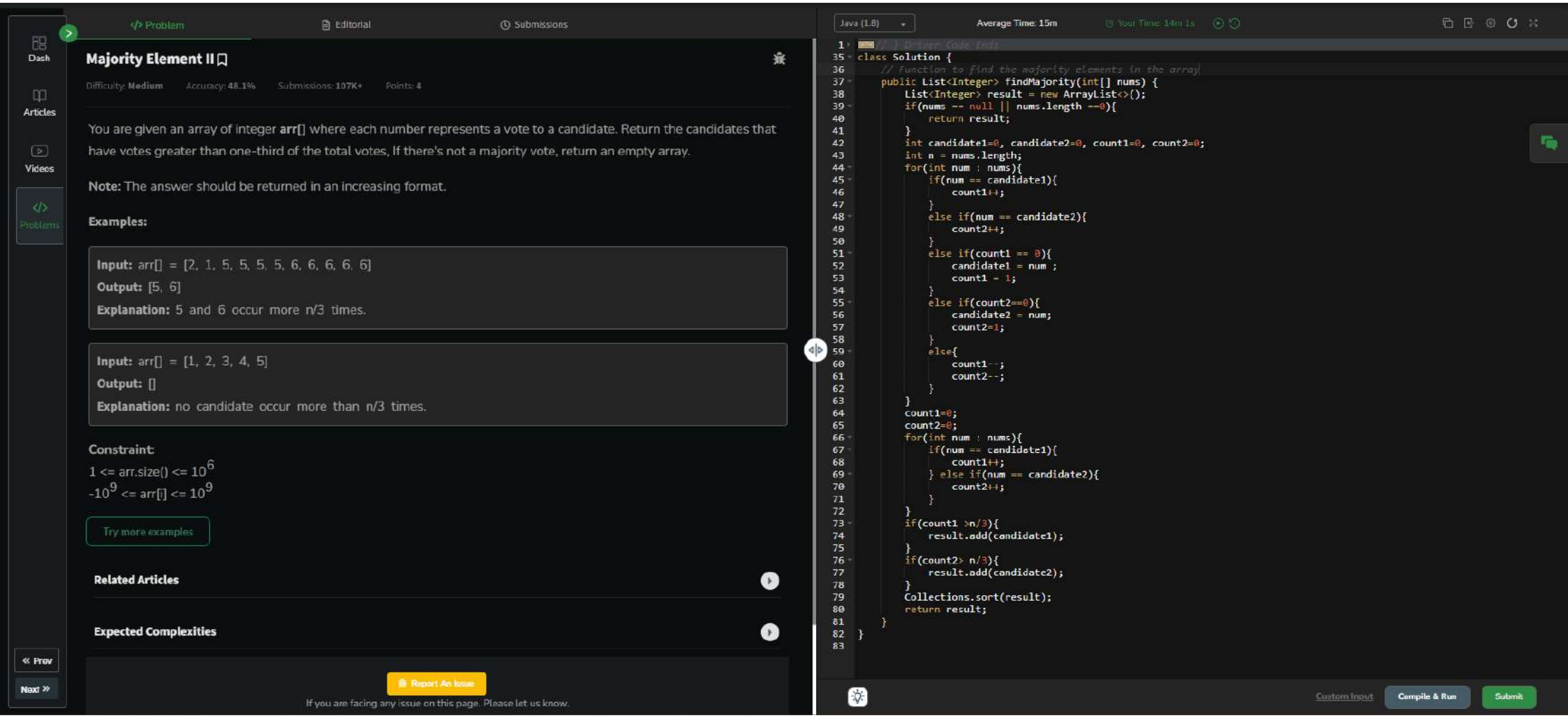
Problem
Editorial
Submissions
Dash
Articles
Videos
Problems
Majority Element II
Difficulty: Medium
Accuracy: 48.1%
Submissions: 107K+
Points: 4
You are given an array of integer arr[] where each number represents a vote to a candidate. Return the candidates that have votes greater than one-third of the total votes, If there's not a majority vote, return an empty array.
Note: The answer should be returned in an increasing format.
Examples:
Input: arr[] = [2, 1, 5, 5, 5, 5, 6, 6, 6, 6, 6]
Output: [5, 6]
Explanation: 5 and 6 occur more n/3 times.
Input: arr[] = [1, 2, 3, 4, 5]
Output: []
Explanation: no candidate occur more than n/3 times.
Constraint:
1 <= arr.size() <= 10^6
-10^9 <= arr[i] <= 10^9
Try more examples
Related Articles
Expected Complexities
Report An Issue
If you are facing any issue on this page. Please let us know.
« Prev
Next »
Java (1.8)
Average Time: 15m
Your Time: 14m 1s
// } Driver Code Ends
class Solution {
// function to find the majority elements in the array
public List<Integer> findMajority(int[] nums) {
List<Integer> result = new ArrayList<>();
if(nums == null || nums.length ==0){
return result;
}
int candidate1=0, candidate2=0, count1=0, count2=0;
int n = nums.length;
for(int num : nums){
if(num == candidate1){
count1++;
}
else if(num == candidate2){
count2++;
}
else if(count1 == 0){
candidate1 = num ;
count1 = 1;
}
else if(count2==0){
candidate2 = num;
count2=1;
}
else{
count1--;
count2--;
}
}
count1=0;
count2=0;
for(int num : nums){
if(num == candidate1){
count1++;
} else if(num == candidate2){
count2++;
}
}
if(count1 >n/3){
result.add(candidate1);
}
if(count2> n/3){
result.add(candidate2);
}
Collections.sort(result);
return result;
}
}
Custom Input
Compile & Run
Submit