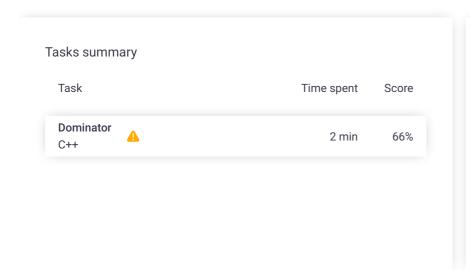
# Codility\_

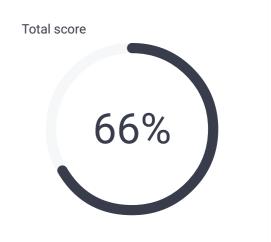
# CodeCheck Report: trainingQCAF5T-G4G

Test Name:

Check out Codility training tasks

Summary Timeline 🖨 Al Assistant Transcript





## **Tasks Details**

## 1. Dominator

Find an index of an array such that its value occurs at more than half of indices in the array.

Task Score

66%

Correctness

50%

Performance

100%

## Task description

An array A consisting of N integers is given. The *dominator* of array A is the value that occurs in more than half of the elements of A.

For example, consider array A such that

$$A[0] = 3$$
  $A[1] = 4$ 

$$A[2] = 3$$

$$A[3] = 2$$
  $A[4] = 3$ 

$$A[5] = -1$$

$$A[6] = 3$$
  $A[7] = 3$ 

The dominator of A is 3 because it occurs in 5 out of 8 elements of A (namely in those with indices 0, 2, 4, 6 and 7) and 5 is more than a half of 8.

Write a function

int solution(vector<int> &A);

that, given an array A consisting of N integers, returns index of any element of array A in which the dominator of A occurs. The function should return -1 if array A does not have a dominator.

For example, given array A such that

$$A[0] = 3$$
  $A[1] = 4$   $A[2] = 3$ 

$$A[3] = 2$$
  $A[4] = 3$   $A[5] = -1$ 

#### Solution

Programming language used: C++

Total time used: 2 minutes

Effective time used: 2 minutes

Notes: not defined yet

Task timeline



Code: 07:16:09 UTC, cpp, show code in pop-up final, score: **66** 

1 // you can use includes, for example:

#include <algorithm>

$$A[6] = 3$$
  $A[7] = 3$ 

the function may return 0, 2, 4, 6 or 7, as explained above.

Write an efficient algorithm for the following assumptions:

- N is an integer within the range [0..100,000];
- each element of array A is an integer within the range [-2,147,483,648..2,147,483,647].

Copyright 2009–2024 by Codility Limited. All Rights Reserved. Unauthorized copying, publication or disclosure prohibited.

#### Test results - Codility

```
3
     #include <map>
5
     // you can write to stdout for debugging purpo
 6
     // cout << "this is a debug message" << endl;</pre>
 7
 8
     int solution(vector<int> &A) {
9
          int answer = -1;
10
          map<int, int> m;
11
12
          for (int i = 0; i < A.size(); i++) {
   if (m.find(A[i]) == m.end())</pre>
13
14
15
                  m.insert(make_pair(A[i], 1));
              else
16
17
                   m[A[i]]++;
18
19
          vector<pair<int, int>> v(m.begin(), m.end
20
21
22
          sort(v.begin(), v.end(), [](pair<int, int>
23
          // for (auto e : v) {
24
25
                 cout << e.first << " " << e.second
          //
          // }
26
27
28
          if (v[0].second != v[1].second){
29
              int leader = v[0].first;
30
31
              for (int i = 0; i < A.size(); i++) {</pre>
32
                   if (A[i] == leader)
33
                       return i;
34
35
          }
36
37
38
          return answer;
39
     }
```

## Analysis summary

The following issues have been detected: wrong answers, runtime errors.

For example, for the input [] the solution terminated unexpectedly.

## **Analysis**

expand all		Example tes	lS
•	example example test		<b>∨</b> OK
ехра	nd all	Correctness to	ests
•	small_nondo	minator all the same elements	<b>∨</b> OK
•	small_half_po half elements th elements the sai	e same, and half + 1	wrong answer got 10, but element is not a dominator, value 2 has only 10 occurences (n=20)
•	small small test		<b>∨</b> OK
•	small_pyrami		<b>∨</b> OK
•	extreme_emp m empty and single		✗ RUNTIME ERROR tested program

## Test results - Codility

rest results - Country						
terminated with exit code						
•	extreme_half1 array with exactly N/2 values 1, N even + [0,0,1,1,1]	•	WRONG ANSWER got 1, but element is not a dominator, value 1 has only 2 occurences (n=4)			
•	extreme_half2 array with exactly floor(N/2) values 1, N odd + [0,0,1,1,1]	×	WRONG ANSWER got 1, but element is not a dominator, value 1 has only 2 occurences (n=5)			
expa	extreme_half3 array with exactly ceil(N/2) values 1 + [0,0,1,1,1] and all Performance to	Ť	oK s			
•	medium_pyramid decreasing and plateau, medium	•	ОК			
•	large_pyramid decreasing and plateau, large		ОК			
•	medium_random random test with dominator, N = 10,000		ОК			
•	large_random random test with dominator, N = 100,000	•	ОК			