Codility_

Screen Report: Anonymous

Test Name:

Check out Codility training tasks

Summary Timeline

Tasks summary

Task	Effective time spent	Score
MissingInteger C++	17 min	100%

Total score

100%

Tasks Details

Medium

1. MissingInteger

Find the smallest positive integer that does not occur in a given sequence.

Task Score	Correctness	Performance
100%	100%	100%

Task description

This is a demo task.

Write a function:

```
int solution(vector<int> &A);
```

that, given an array A of N integers, returns the smallest positive integer (greater than 0) that does not occur in A.

For example, given A = [1, 3, 6, 4, 1, 2], the function should return 5.

Given A = [1, 2, 3], the function should return 4.

Given A = [−1, −3], the function should return 1.

Write an **efficient** algorithm for the following assumptions:

- N is an integer within the range [1..100,000];
- each element of array A is an integer within the

Solution

Programming language used:	C++
Time spent on task:	17 minutes
Notes:	*not defined yet*

Task timeline

15:02:18 15:18:54

Code: 15:18:53 UTC, cpp, show code in pop-up

range [−1,000,000..1,000,000].

final, score: **100**

```
1  // you can use includes, for example:
2  // #include <algorithm>
3
4  // you can write to stdout for debugging purposes,
5  // cout << "this is a debug message" << endl;
6
7  int solution(vector<int> &A) {
8      // Implement your solution here
9
10     // create a tracker vector to count which inte
11     int sizeA{(int) A.size()};
12     int counterSize{sizeA + 1}; // has extra eleme
13     vector<bool> counter(counterSize, false);
14
15     for(auto num: A)
16     {
17         if(num > 0 && num < counterSize)
18         {
19             counter[num] = true;
20         }
21     }
22
23     for(int i = 1; i < counterSize; i++)
24     {
25         if(counter[i] == false)
26         {
27             return i;
28         }
29     }
30
31     return counterSize;
32 }
```

Analysis summary

The solution obtained perfect score.

Analysis

Detected time complexity: **O(N) or O(N * log(N))**

expand all	Example tests	
▶	example1 first example test	✓ OK
▶	example2 second example test	✓ OK
▶	example3 third example test	✓ OK
expand all	**Correctness tests**	
▶	extreme_single a single element	✓ OK
▶	simple	✓ OK

simple test

▶	extreme_min_max_value minimal and maximal values	✓ OK
▶	positive_only shuffled sequence of 0...100 and then 102...200	✓ OK
▶	negative_only shuffled sequence -100 ... -1	✓ OK

expand all **Performance tests**

▶	medium chaotic sequences length=10005 (with minus)	✓ OK
▶	large_1 chaotic + sequence 1, 2, ..., 40000 (without minus)	✓ OK
▶	large_2 shuffled sequence 1, 2, ..., 100000 (without minus)	✓ OK
▶	large_3 chaotic + many -1, 1, 2, 3 (with minus)	✓ OK