codility

Training center

Check out Codility training tasks

Candidate Report: Anonymous

Test Name:

SUMMARY

Test Score

100 out of 100 points

100%

Tasks in Test

	Time Spent	Task Score
MissingInte... Submitted in: C++	1 min	100%

TASKS DETAILS

1. MissingInteger

EASY

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.

Solution

Programming language used:	C++
Total time used:	1 minutes
Effective time used:	1 minutes
Notes:	*not defined yet*

Task timeline

Assume that:

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

Complexity:

- expected worst-case time complexity is O(N);
- expected worst-case space complexity is O(N) (not counting the storage required for input arguments).

13:31:59 13:32:14

Code: 13:32:14 UTC, cpp, final, score: **100** show code in pop-up

```cpp
#include <iostream>
#include <vector>
#include <algorithm>

using namespace std;
typedef int INT64;
typedef vector<INT64> vINT64;


INT64 solution(vINT64 &vInput)
{

        sort(vInput.begin(), vInput.end());
        //vInput.erase(unique(vInput.begin(), vI

        INT64 hi = vInput[vInput.size()-1];
        if (hi <= 0)
                return 1;

        vINT64::iterator it = vInput.begin();

        INT64 now = 1;

        while (it != vInput.end())
        {
                if (*it < 0)
                        it++;
                else
                {
                        if (*it < now)
                                it++;
                        else if (*it == now)
                        {
                                it++;
                                now++;
                        }
                        else
                                return now;
                }

        }



        return hi +1;
}
```

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 simple test		✓ OK
▸ 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