codility

Check out Codility training tasks

Candidate Report: Anonymous

Test Name:

Summary Timeline

Test Score

100 out of 100 points

100%

Tasks in Test

	Time Spent	Task Score
MissingInteger Submitted in: C++	1 min	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.

Solution

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

Task timeline

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 range [−1,000,000..1,000,000].

06:15:23 06:15:48

Code: 06:15:48 UTC, cpp, final, score: **100** show code in pop-up

```cpp
// you can use includes, for example:
#include <algorithm>
#include <vector>
#include <unordered_set>

// you can write to stdout for debugging purpose
// cout << "this is a debug message" << endl;

int solution(std::vector<int>& A) {
    // write your code in C++14 (g++ 6.2.0)
    int result = 1;

    // Remove duplicates and negatives
    std::unordered_set<int> positive_int_set;
    for (int value : A) {
        if (value > 0) {
            positive_int_set.insert(value);
        }
    }
    std::vector<int> positive_ints;
    positive_ints.reserve(positive_int_set.size()
    positive_ints.assign(positive_int_set.begin()

    // Sort the values
    std::sort(positive_ints.begin(), positive_int

    // Now we have only the positive integers
    for (auto value : positive_ints) {
        if (value != result) {
            break;
        }
        ++result;
    }
    return result;
}
```

Analysis summary

The solution obtained perfect score.

Analysis

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

collapse all **Example tests**

▼ example1 ✓ OK
first example test

1. 0.001 s OK

▼ example2 ✓ OK
second example test

1. 0.001 s OK

▼ example3 ✓ OK
third example test

1. 0.001 s OK

collapse all **Correctness tests**

▼ extreme_single ✓ OK
a single element

1. 0.001 s OK
2. 0.001 s OK
3. 0.001 s OK
4. 0.001 s OK

▼ simple ✓ OK
simple test

1. 0.001 s OK
2. 0.001 s OK
3. 0.001 s OK

▼ extreme_min_max_value ✓ OK
minimal and maximal values

1. 0.001 s OK
2. 0.001 s OK

▼ positive_only ✓ OK
shuffled sequence of 0...100 and then 102...200

1. 0.001 s OK
2. 0.001 s OK

▼ negative_only ✓ OK
shuffled sequence -100 ... -1

1. 0.001 s OK

collapse all

Performance tests

▼ medium ✓ OK
chaotic sequences length=10005 (with minus)

1. 0.001 s OK
2. 0.001 s OK
3. 0.004 s OK

▼ large_1 ✓ OK
chaotic + sequence 1, 2, ..., 40000 (without minus)

1. 0.020 s OK

▼ large_2 ✓ OK
shuffled sequence 1, 2, ..., 100000 (without minus)

1. 0.028 s OK
2. 0.028 s OK

▼ large_3 ✓ OK
chaotic + many -1, 1, 2, 3 (with minus)

1. 0.008 s OK

PDF version of this report that may be downloaded on top of this site may contain sensitive data including personal information. For security purposes, we recommend you remove it from your system once reviewed.