codility

Check out Codility training tasks

Candidate Report: Vadim Tsozik

Test Name:

Summary Timeline

Test Score

100 out of 100 points

100%

Tasks in Test

	Time Spent	Task Score
MissingInteger Submitted in: C++	2 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:

Solution

Programming language used: C++

```
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.

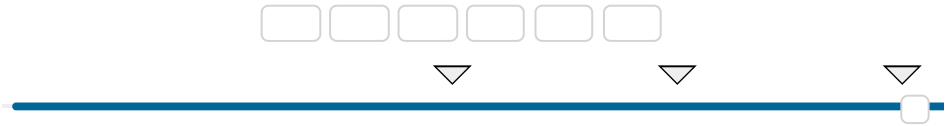
Given A = [−1, −3], the function should return 1.

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 range [−1,000,000..1,000,000].

Total time used:	2 minutes
Effective time used:	2 minutes
Notes:	*not defined yet*

Task timeline



01:16:50 01:18:21

Code: 01:18:21 UTC, cpp, final, score: **100** show code in pop-up

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

// you can write to stdout for debugging purposes, e.g.
// cout << "this is a debug message" << endl;
#include <unordered_set>

int solution(vector<int> &A) {
    // write your code in C++14 (g++ 6.2.0)
    if(A.empty())
        return 1;
    unordered_set<int> s;
    int max(0);
    for (size_t i(0); i < A.size(); ++i)
    {
        if(A[i] > 0) {
            if(A[i] > max) {
                max = A[i];
            }
            s.insert(A[i]);
        }
    }
    int t = 1;
```

```
24      for(; t <= max; ++t)
25      {
26          if(s.find(t) == s.end())
27              return t;
28      }
29      return t;
30
31  }
```

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