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:

```
class Solution { public int solution(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].

Solution

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

Task timeline

18:21:54 18:22:23

Code: 18:22:23 UTC, cs, final, score: **100** show code in pop-up

```
1  using System;
2  // you can also use other imports, for example:
```

```
3  // using System.Collections.Generic;
4
5  // you can write to stdout for debugging purposes, e.g.
6  // Console.WriteLine("this is a debug message");
7
8  class Solution {
9      public int solution(int[] A) {
10         // write your code in C# 6.0 with .NET 4.5 (Mono)
11
12  var result = 1;
13
14             Array.Sort(A);
15
16             for (int i = 1; i <= 100000; i++)
17             {
18                 if (Array.BinarySearch(A, i) < 0)
19                 {
20                     result = i;
21                     break;
22                 }
23
24                 result = 100001;
25             }
26
27             return result;
28     }
29 }
```

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	