codility

Check out Codility training tasks

Candidate Report: Anonymous

Test Name:

Summary Timeline

Test Score

88 out of 100 points

88%

Tasks in Test

	Time Spent	Task Score
MissingInteger Submitted in: JavaScript	10 min	88%

TASKS DETAILS

MEDIUM

1. MissingInteger

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

Task Score	Correctness	Performance
88%	80%	100%

Task description

This is a demo task.

Write a function:

```
function solution(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:	JavaScript
Total time used:	10 minutes
Effective time used:	10 minutes
Notes:	*not defined yet*

Task timeline

15:24:04 15:33:34

Code: 15:33:33 UTC, js, final, score: **88** show code in pop-up

```
1  // you can write to stdout for debugging purpose
2  // console.log('this is a debug message');
3
4  function solution(A) {
5    const arraySet = new Set(A);
6    const uniqueItemsArray = [...arraySet];
7    const positiveIntegersArray = [];
8    let missingInteger = null;
9
10   for (let i = 0; i < uniqueItemsArray.length; i
11     if (uniqueItemsArray[i] > 0) {
12       positiveIntegersArray.push(uniqueItemsArra
13     }
14   }
15
16   const sortedArray = positiveIntegersArray.sort
17
18   if (sortedArray.length === 0 || sortedArray[0]
19     missingInteger = 1;
20   } else {
21     for (let i = 1; i < sortedArray.length; i +=
22       if (sortedArray[i] - sortedArray[i - 1] !=
23         missingInteger = sortedArray[i - 1] + 1;
24         break;
25       } else {
26         missingInteger = sortedArray[i] + 1;
27       }
28     }
29   }
30
31   return missingInteger;
32 }
```

Analysis summary

The following issues have been detected: runtime errors.

For example, for the input `[1]` the solution terminated unexpectedly.

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	✗ RUNTIME ERROR tested program terminated with exit code 1
▶	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