Codility_

CodeCheck Report: trainingRCGDJV-JVX

Test Name:

Check out Codility training tasks

Summary Timeline

Tasks summary

Task	Time spent	Score
MissingInteger Python	1 min	100%

Total score

Tasks Details

1. **MissingInteger**

Medium

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:

```
def 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.

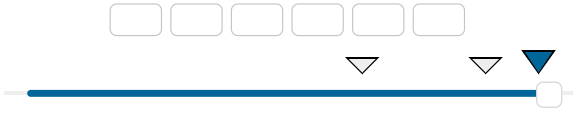
Given A = [−1, −3], the function should return 1.

Write an **efficient** algorithm for the following assumptions:

Solution

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

Task timeline



- 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].

13:08:29 13:09:00

Code: 13:08:59 UTC, py, final, score: **100** show code in pop-up

```
1  # you can write to stdout for debugging p
2  # print("this is a debug message")
3
4  def solution(A):
5      # Implement your solution here
6      # pass
7      sorted_set_A = sorted(set(filter(lam
8
9      for i, element in enumerate(sorted_s
10         if element != i + 1:
11             return i + 1
12
13     return len(sorted_set_A) + 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
collapse all	**Performance tests**	
▼	medium chaotic sequences length=10005 (with minus)	✔ OK

1. 0.016 s **OK**
2. 0.016 s **OK**
3. 0.016 s **OK**

▼ large_1 ✔ **OK**

chaotic + sequence 1, 2, ..., 40000 (without minus)

1. 0.060 s **OK**

▼ large_2 ✔ **OK**

shuffled sequence 1, 2, ..., 100000 (without minus)

1. 0.068 s **OK**
2. 0.064 s **OK**

▼ large_3 ✔ **OK**

chaotic + many -1, 1, 2, 3 (with minus)

1. 0.056 s **OK**