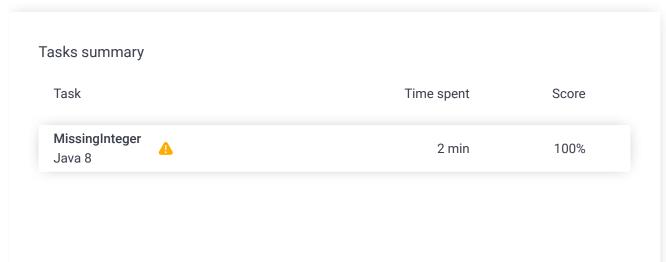
Codility_

Candidate Report: Anonymous

Check out Codility training tasks

Test Name:

Summary Timeline





Tasks Details

1. MissingInteger Task Score Correctness Performance
Find the smallest positive integer that does not occur in a given

sequence. 100% 100% 100%

Task description Solution

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].

Copyright 2009–2021 by Codility Limited. All Rights Reserved. Unauthorized copying, publication or disclosure prohibited.

Programming language used: Java 8

Total time used: 2 minutes

Effective time used: 2 minutes

Notes: not defined yet

Task timeline

12:50:08

Code: 12:51:47 UTC, java, final, score: show code in pop-up 100

```
import java.util.*;
     class Solution {
       public int solution(int[] A) {
         List<Integer> targetList = new ArrayList<>(A.length);
 4
         for (int i : A) {
 6
           targetList.add(i);
 7
         Collections.sort(targetList);
 8
 9
         if ((targetList.get(targetList.size() - 1) < 1) ||</pre>
10
             (targetList.get(0) > 1)) {
11
           return 1;
12
13
         Iterator<Integer> it = targetList.iterator();
14
         int first = it.next();
15
         int previousPositive = first > 0 ? first : 0;
         while (it.hasNext()) {
16
           int current = it.next();
17
           if (current > 0) {
18
             if ((current > 1) && (previousPositive < 1)) {</pre>
19
20
               return 1;
21
             if (current - previousPositive > 1) {
22
23
               return previousPositive + 1;
```

24	}
25	<pre>previousPositive = current;</pre>
26	}
27	}
28	return previousPositive + 1;
29	}
30	}

Analysis summary

The solution obtained perfect score.

Analysis

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

expan	d 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		✓	ОК
•	positive_only shuffled sequence of 0100 and	d then 102200	✓	OK
•	negative_only shuffled sequence -1001		✓	OK

expand	d all Performance tes	ts
•	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

The 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.