

## Candidate Report: Alberto Petazzi

Test Name:

[Summary](#) [Timeline](#)

Mail Status: Not Applicable

User Note

Finished: 2020-02-01 15:34 UTC

Started: 2020-02-01 15:34 UTC

You can add the note here.

### 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].

Copyright 2009–2020 by Codility Limited. All Rights Reserved. Unauthorized copying, publication or disclosure prohibited.

### Solution

Programming language used: Java 8

Total time used: 1 minutes

?

Effective time used: 1 minutes

?

Notes: *not defined yet*

### Task timeline

?



15:34:20

15:34:51

Code: 15:34:51 UTC, java, final,  
score: 100[show code in pop-up](#)

```
1 // you can also use imports, for example:
2 // import java.util.*;
3
4 // you can write to stdout for debugging purposes, e.g.
5 // System.out.println("this is a debug message");
6
7 class Solution {
8     public static int max = 1000001;
9     public static boolean[] mem = new boolean[max];
10
11     public static int solution(int[] a) {
12
```

```
13         for (int i = 0; i < a.length; i++)
14             if (a[i] > 0)
15                 mem[a[i]] = true;
16
17         int i = 1;
18         while (mem[i])
19             i++;
20
21         return i;
22     }
23 }
```

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

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.