Codility_

CodeCheck Report: trainingESTRS7-A8N

Test Name:

Check out Codility training tasks

Summary Timeline

Tasks summary

Task	Time spent	Score
MissingInteger ⚠ Java 8	33 min	100%

Total score

100%

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:

```
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.

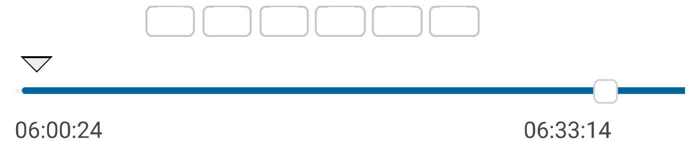
Write an **efficient** algorithm for the following assumptions:

- 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:	Java 8
Total time used:	33 minutes
Effective time used:	33 minutes
Notes:	*not defined yet*

Task timeline



Code: 06:33:14 UTC, java, final, score: **100**

show code in pop-up

```
// you can also use imports, for example:
// import java.util.*;

// you can write to stdout for debugging purposes,
// System.out.println("this is a debug message");

class Solution {
        public int solution(int[] A) {
        int A_N = A.length;
        boolean [] B = new boolean[A_N + 1];
        int B_N = B.length;
        for (int i = 0; i < A_N; i++) {

                if (A[i] > 0 && A[i] < B_N) {
                        B[A[i]] = true;
                }
        }
        for (int i = 1; i < B_N ; i++) {
                if (B[i] == false) {
                        return i;
                }
        }
        return B_N;
    }
}
```

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