codility

Congratulations

You have completed a Codility training test.

Tweet this!
I scored 100% in #java on @Codility!
https://codility.com/demo/take-sample-test/missing_integer

Sign up for our newsletter!

Like us on Facebook!

Training ticket

Session

ID: trainingBQ9RC5-2G5
Time limit: 120 min.

Status: closed

Created on: 2016-06-20 20:12 UTC
Started on: 2016-06-20 20:12 UTC
Finished on: 2016-06-20 20:25 UTC

Tasks in test	Correctness	Performance	Task score
1 MissingInteger Submitted in: Java	100%	100%	100%

Test score

100%

100 out of 100 points

1. MissingInteger

EASY

Find the minimal positive integer not occurring in a given sequence.

score: 100 of 100

Task description

Write a function:

```
class Solution { public int solution(int[] A); }
```

that, given a non-empty zero-indexed array A of N integers, returns the minimal positive integer (greater than 0) that does not occur in A.

For example, given:

```
A[0] = 1
A[1] = 3
A[2] = 6
A[3] = 4
A[4] = 1
A[5] = 2
```

the function should return 5.

Assume that:

- N is an integer within the range [1..100,000];
- each element of array A is an integer within the range [−2,147,483,648..2,147,483,647].

Complexity:

- expected worst-case time complexity is O(N);
- expected worst-case space complexity is O(N), beyond input storage (not counting the storage required for input arguments).

Elements of input arrays can be modified.

Solution

Programming language used: Java

Total time used: 13 minutes

Effective time used: 13 minutes

Notes: *not defined yet*

Task timeline

20:12:53 20:25:53

Code: 20:25:53 UTC, java, final, score: 100

show code in pop-up

```
import java.util.List;
import java.util.stream.IntStream;

import static java.util.stream.Collectors.toList;

class Solution {
    public int solution(int[] A) {
        List<Integer> collect = IntStream.of(A)
                .filter(value -> value > 0)
                .distinct()
                .sorted()
                .boxed()
                .collect(toList());

        for (int i = 0; i < collect.size(); i++) {
            if(collect.get(i) != i+1) {
                return i+1;
            }
        }
        return collect.size()+1;
    }
}
```

Analysis summary

The solution obtained perfect score.

Analysis

Detected time complexity:
O(N)

expand all **Example tests**

▶ example example (without minus)	✔ OK

expand all **Correctness tests**

▶ extreme_single a single element	✔ OK
▶ simple simple test	✔ OK
▶ extreme_min_max_int MININT and MAXINT (with minus)	✔ 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

Training center