Codility_

Candidate Report: trainingHCM3Q7-9X8

Check out Codility training tasks

Test Name:

Summary | Timeline

Tasks summary

Task	Time spent	Score
PermMissingElem Java 8	23 min	100%

Total score

100%

Tasks Details

Easy

1. PermMissingElem
Find the missing element in a given permutation.

Task Score	Correctness	Performance
100%	100%	100%

Task description

An array A consisting of N different integers is given. The array contains integers in the range [1..(N + 1)], which means that exactly one element is missing.

Your goal is to find that missing element.

Write a function:

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

that, given an array A, returns the value of the missing element.

For example, given array A such that:

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

the function should return 4, as it is the missing element.

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

- N is an integer within the range [0..100,000];
- the elements of A are all distinct;
- each element of array A is an integer within the range [1..(N + 1)].

Solution

Programming language used:	Java 8
Total time used:	23 minutes
Effective time used:	23 minutes
Notes:	*not defined yet*

Task timeline

21:12:57 — 21:35:37

Code: 21:35:37 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 int solution(int[] A) {
9          // write your code in Java SE 8
10         int len = A.length;
11         if(len == 0) return 1;
12
13         int sumA = 0;
14         for(int i=0; i < len; i++){
15             sumA += A[i];
16         }
17
18
19         int sumI = 0;
20         for(int i=1;i <= len +1 ; i++){
21             sumI += i;
22         }
23
24         return  sumI - sumA;
25     }
26 }
```

Analysis summary

The solution obtained perfect score.

Analysis

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

expand all	Example tests	
▶	example example test	✔ OK
expand all	**Correctness tests**	
▶	empty_and_single empty list and single element	✔ OK
▶	missing_first_or_last the first or the last element is missing	✔ OK
▶	single single element	✔ OK
▶	double two elements	✔ OK
▶	simple simple test	✔ OK
expand all	**Performance tests**	
▶	medium1 medium test, length = ~10,000	✔ OK
▶	medium2 medium test, length = ~10,000	✔ OK
▶	large_range range sequence, length = ~100,000	✔ OK
▶	large1 large test, length = ~100,000	✔ OK
▶	large2 large test, length = ~100,000	✔ OK