Codility_

CodeCheck Report: training98BBNR-932

Test Name:

Check out Codility training tasks

Summary Timeline

Tasks summary

Task	Time spent	Score
PermMissingElem ⚠ Python	3 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:

```
def solution(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:	Python
Total time used:	3 minutes
Effective time used:	3 minutes
Notes:	*not defined yet*

Task timeline

00:08:37 00:10:44

Code: 00:10:44 UTC, py, final, score: 100

show code in pop-up

```
# you can write to stdout for debugging purposes, e.g.
# print("this is a debug message")

def solution(A):
    # write your code in Python 3.6
    if len(A) == 0:
        return 1
    elif len(A) == 1:
        if A[0] == 1:
            return 2
        else:
            return 1
    else:
        B = [0] * (len(A)+1)
        for el in A:
            B[el-1] = 1
        for i in range(len(B)+1):
            if i > 0:
                if B[i-1] == 0:
                    return i
```

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