Codility_

Candidate Report: trainingCZ8EVT-BF6

Check out Codility training tasks

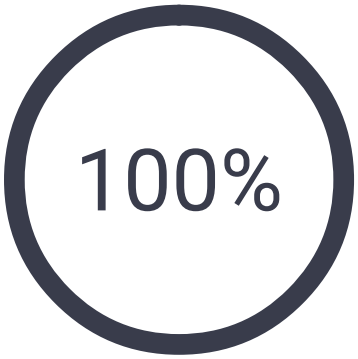
Test Name:

Summary Timeline

Tasks summary

Task	Time spent	Score
PermMissingElem Python	4 min	100%

Total score



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];

Solution

Programming language used:	Python
Total time used:	4 minutes
Effective time used:	4 minutes
Notes:	*not defined yet*

Task timeline

09:56:39 10:00:00

Code: 10:00:00 UTC, py, final, score: **100** show code in pop-up

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


```python
# 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
    dic = {}
    for i in A:
        dic[i] = 1
    i = 1
    while True:
        if i not in dic:
            return i
        i = i + 1
    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

The 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.