Codility_

Candidate Report: trainingC2Q7EF-GUY

Check out Codility training tasks

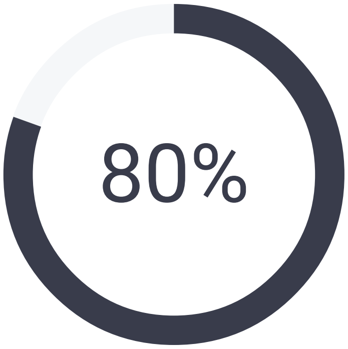
Test Name:

Summary Timeline

Tasks summary

Task	Time spent	Score
PermMissingElem ⚠ Java 8	21 min	80%

Total score



Tasks Details

Easy

1. PermMissingElem

Find the missing element in a given permutation.

Task Score	Correctness	Performance
80%	100%	60%

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:	21 minutes
Effective time used:	21 minutes
Notes:	*not defined yet*

Task timeline

20:48:03

21:08:49

Code: 21:08:49 UTC, java, final, score: **80**

show code in pop-up

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

```

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

class Solution {
    public int solution(int[] A) {
        // write your code in Java SE 8
        if(A.length == 0) return 1;

        int sumA = 0;
        for(int i=0; i < A.length; i++){
            sumA += A[i];
        }
        int arraylen = A.length + 1;
        int actualSum = (arraylen * (arraylen + 1)) / 2

        return actualSum - sumA;
    }
}
```

Analysis summary

The following issues have been detected: wrong answers.

Analysis

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

expand all **Example tests**

▶ example ✔ OK
example test

expand all **Correctness tests**

▶ empty_and_single ✔ OK
empty list and single element

▶ missing_first_or_last ✔ OK
the first or the last element is missing

▶ single ✔ OK
single element

▶ double ✔ OK
two elements

▶ simple ✔ OK
simple test

expand all **Performance tests**

▶ medium1 ✔ OK
medium test, length = ~10,000

▶ medium2 ✔ OK
medium test, length = ~10,000

▼ large_range ✘ WRONG ANSWER
range sequence, length = ~100,000 got -2147483647 expected 1

1. 0.244 s **OK**
2. 0.124 s **WRONG ANSWER**, got -2147483647 expected 1
3. 0.124 s **WRONG ANSWER**, got -2147471303 expected 12345

▶ large1 ✔ OK
large test, length = ~100,000

▼ large2 ✘ WRONG ANSWER
large test, length = ~100,000 got -2147473647 expected

10001

1. 0.148 s **WRONG ANSWER**, got -2147473647 expected 10001

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.