

Candidate Report: Anonymous

Test Name:

[Summary](#)[Timeline](#)

Test Score

100 out of 100 points

100%

Tasks in Test

PermMissingElem
Submitted in: Java 8

Time Spent ⓘ

1 min

Task Score

100%

TASKS DETAILS

EASY

1. PermMissingElem

Find the missing element in a given permutation.

Task Score

Correctness

100%

Performance

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 $[1..(N + 1)]$.

Copyright 2009–2020 by Codility Limited. All Rights Reserved. Unauthorized copying, publication or disclosure prohibited.

Solution

Programming language used: Java 8

Total time used: 1 minutes ⓘ

Effective time used: 1 minutes ⓘ

Notes: *not defined yet*

Task timeline

 ⓘ

10:02:45

10:03:22

Code: 10:03:22 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 static int max = 1000000;
9     public static boolean[] mem;
```

```
10
11
12     public static int solution(int[] a) {
13         mem = new boolean[max + 1];
14         if (a.length == 0)
15             return 1;
16
17         for (int i = 0; i < a.length; i++) {
18             mem[a[i]] = true;
19         }
20
21         for (int i = 1; i < a.length + 1; i++) {
22             if (!mem[i])
23                 return i;
24         }
25
26         return a.length + 1;
27     }
28 }
```

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