

Tasks summary

Task	Time spent	Score
PermMissingElem Java 8	1 min	100%

Total score

100%

Tasks Details

Easy	1. PermMissingElem	Task Score	Correctness	Performance	
	Find the missing element in a given permutation.		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:

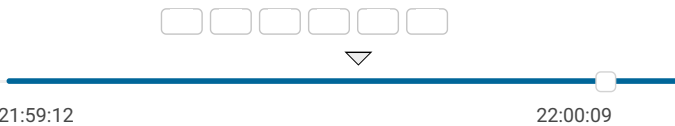
- N is an integer within the range [0..100,000];
- the elements of A are all distinct;
- each element of array A is an integer within the range [1..(N + 1)].

Copyright 2009–2023 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



Code: 22:00:09 UTC, java, final, score: 100

show code in pop-up

```
1 import java.util.BitSet;
2
3 class Solution {
4     public int solution(int[] A) {
5         BitSet B = new BitSet(A.length + 1);
6         for (int i = 0; i < A.length; i++) {
7             B.set(A[i] - 1);
8         }
9         return B.nextClearBit(0) + 1;
10    }
11 }
```

Analysis summary

The solution obtained perfect score.

Analysis

Detected time complexity:

O(N) or O(N * log(N))

collapse all	Example tests	
▼ example		✓ OK
example test		

1.	0.004 s	OK
collapse all	Correctness tests	
▼ empty_and_single		✓ OK
empty list and single element		

1.	0.004 s	OK
2.	0.004 s	OK
▼ missing_first_or_last		✓ OK
the first or the last element is missing		

1.	0.004 s	OK
2.	0.004 s	OK
▼ single		✓ OK
single element		

1.	0.004 s	OK
2.	0.004 s	OK
▼ double		✓ OK
two elements		

1.	0.004 s	OK
2.	0.004 s	OK
3.	0.004 s	OK
▼ simple		✓ OK
simple test		

1.	0.004 s	OK
collapse all	Performance tests	
▼ medium1		✓ OK
medium test, length = ~10,000		

1.	0.032 s	OK
▼ medium2		✓ OK
medium test, length = ~10,000		

1.	0.032 s	OK
▼ large_range		✓ OK
range sequence, length = ~100,000		

1.	0.312 s	OK
2.	0.160 s	OK
3.	0.160 s	OK
▼ large1		✓ OK
large test, length = ~100,000		

1.	0.320 s	OK
▼ large2		✓ OK
large test, length = ~100,000		

1.	0.192 s	OK