closed

Demo ticket

Session
ID: demoJC5H7H-367
Time limit: 120 min.

Status: closed
Started on: 2014-01-05 06:53 UTC

Score:

100

of 100

1. PermMissingElem

Find the missing element in a given permutation.

score: 100 of 100

Task description

A zero-indexed 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 a zero-indexed 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.
Assume that:

- 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)].

Complexity:

- expected worst-case time complexity is O(N);
- expected worst-case space complexity is O(1), beyond input storage (not counting the storage required for input arguments).

Elements of input arrays can be modified.

Solution

Programming language used: C#

Total time used: 1 minutes (?)

Effective time used: 1 minutes (?)

Notes: correct functionality and scalability

Task timeline What is it?

06:53:43 06:54:38

Code: 06:54:38 UTC, cs, final, score: 100.00

```
01. using System;
02. // you can also use other imports, for
       example:
03. // using System.Collections.Generic;
04. class Solution {
05.         public int solution(int[] A)
06.         {
07.             if (A == null) throw new
                   ArgumentNullException();
08.             if (A.Length > 100000) throw
                   new
                   ArgumentOutOfRangeException();
09.             // write your code in C# with
                   .NET 2.0
10.             var values = new bool[A.Length
                   + 1];
11.
12.             var result = 0;
13.             foreach (var value in A)
14.                 values[value - 1] = true;
15.
16.             for (var index = 0; index <
                   values.Length; index++)
17.             {
18.                 if (!values[index])
19.                 {
20.                     result = index + 1;
21.                     break;
22.                 }
23.             }
24.
25.             return result;
26.         }
27. }
```

Analysis

Detected time complexity:

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

test	time	result
example example test	0.080 s.	**OK**
empty empty list	0.080 s.	**OK**
single single element	0.080 s.	**OK**
double two elements	0.080 s.	**OK**
simple simple test	0.080 s.	**OK**
medium1 medium test, length = ~10,000	0.080 s.	**OK**
medium2 medium test, length = ~10,000	0.080 s.	**OK**
large_range range sequence, length = ~100,000	0.090 s.	**OK**
large1 large test, length = ~100,000	0.100 s.	**OK**
large2 large test, length = ~100,000	0.090 s.	**OK**

Get acc