closed

Demo ticket

Session
ID: demo6HHBMN-CK8
Time limit: 120 min.

Status: closed
Started on: 2014-01-05 06:48 UTC

Score:

100

of 100

1. PermCheck

Check whether array A is a permutation.

score: 100 of 100

Task description

A non-empty zero-indexed array A consisting of N integers is given.
A *permutation* is a sequence containing each element from 1 to N once, and only once.
For example, array A such that:

```
A[0] = 4
A[1] = 1
A[2] = 3
A[3] = 2
```

is a permutation, but array A such that:

```
A[0] = 4
A[1] = 1
A[2] = 3
```

is not a permutation.
The goal is to check whether array A is a permutation.
Write a function:

```
class Solution { public int solution(int[] A); }
```

that, given a zero-indexed array A, returns 1 if array A is a permutation and 0 if it is not.
For example, given array A such that:

```
A[0] = 4
A[1] = 1
A[2] = 3
A[3] = 2
```

the function should return 1.
Given array A such that:

```
A[0] = 4
A[1] = 1
A[2] = 3
```

the function should return 0.
Assume that:

- N is an integer within the range [1..100,000];
- each element of array A is an integer within the range [1..1,000,000,000].

Complexity:

- expected worst-case time complexity is O(N);
- expected worst-case space complexity is O(N), 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: 2 minutes (?)

Effective time used: 1 minutes (?)

Notes: correct functionality and scalability

Task timeline What is it?

06:48:05 06:49:11

Code: 06:49:11 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.             int[] numbers = new
                   int[A.Length];
08.             foreach (var value in A)
09.             {
10.                 if (value > A.Length)
11.                     return 0;
12.
13.                 numbers[value - 1]++;
14.             }
15.
16.             foreach (var value in numbers)
17.             {
18.                 if (value != 1) return 0;
19.             }
20.             return 1;
21.         }
22. }
```

Analysis

Detected time complexity:

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

test	time	result
example1 the first example test	0.080 s.	**OK**

example2 the second example test	0.080 s.	**OK**
extreme_max single element with maximal value	0.080 s.	**OK**
single single element	0.070 s.	**OK**
double two elements	0.080 s.	**OK**
antiSum1 total sum is corret (equals 1 + 2 + ... N), but it is not a permutation, N = 3	0.080 s.	**OK**
medium_permutation permutation, N = ~10,000	0.080 s.	**OK**
antiSum2 total sum is corret (equals 1 + 2 + ... N), but it is not a permutation, N = ~100,000	0.110 s.	**OK**
large_permutation large permutation, N = ~100,000	0.100 s.	**OK**
large_range sequence 1, 2, ..., N, N = ~100,000	0.110 s.	**OK**
extreme_values all the same values, N = ~100,000	0.100 s.	**OK**

Get acc