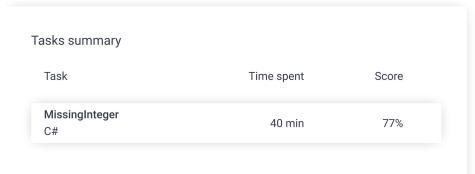
Codility_

Candidate Report: trainingPJ43Z5-XKA

Check out Codility training tasks

Test Name:

Feedback Summary Timeline





Tasks Details

1. MissingInteger Task Score Correctness Performance Find the smallest positive integer that does 100% 77% 50% not occur in a given sequence.

Task description

This is a demo task.

Write a function:

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

that, given an array A of N integers, returns the smallest positive integer (greater than 0) that does not occur in A.

For example, given A = [1, 3, 6, 4, 1, 2], the function should return 5.

Given A = [1, 2, 3], the function should return 4.

Given A = [-1, -3], the function should return 1.

Write an efficient algorithm for the following assumptions:

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

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

Solution

Programming language used: Total time used: 40 minutes Effective time used: 40 minutes Notes: not defined yet Task timeline 01:50:25 02:29:58 Code: 02:29:58 UTC, cs, final, show code in pop-up score: 77

```
using System;
     using System.Linq;
     // you can also use other imports, for example:
3
     // using System.Collections.Generic;
     // you can write to stdout for debugging purposes, e.g.
6
     // Console.WriteLine("this is a debug message");
8
9
     class Solution {
             bool isContiguous(int []arr)
10
11
12
                 int cnt = arr.Count();
13
                 var max = arr.Max();
14
                 var min = arr.Min();
                 if (max - min == cnt - 1)
15
16
                      return true;
                 else
17
18
                      return false; ;
19
             }
20
21
             public int solution(int[] A)
22
23
                 if (A.Count(x=>x>0) == 0)
24
                 {
25
                      return 1;
26
                  }else
27
                 {
28
                      var positiveSubset = A.Where(x \Rightarrow x \Rightarrow 1).C
                      if (positiveSubset.Min() > 1)
29
30
                          return 1;
31
                      if (isContiguous(positiveSubset.ToArray()))
32
                          return positiveSubset.Max() + 1;
33
                      else
34
35
                          positiveSubset = positiveSubset.OrderBy
36
                          var fullArray = Enumerable.Range(1, A.M)
                          return fullArray.Except(positiveSubset)
37
38
                     }
39
                 }
             }
40
41
     }
```

Analysis summary

The following issues have been detected: timeout errors.

Analysis 2

Detected time complexity:

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

expand all		Example tests	Example tests		
•	example1 first example test	✓	ОК		
•	example2 second example test	✓	ОК		
•	example3 third example test	✓	ОК		
expand all		Correctness tests			
•	extreme_single a single element	✓	ОК		

•	simple simple test	✓	ОК			
•	extreme_min_max_value minimal and maximal values	√	OK			
•	positive_only shuffled sequence of 0100 and then 102200	✓	ОК			
•	negative_only shuffled sequence -1001	✓	OK			
expand all Performance tests						
•	medium chaotic sequences length=10005 (with minus)	X	TIMEOUT ERROR running time: 0.412 sec., time limit: 0.100 sec.			
•	large_1 chaotic + sequence 1, 2,, 40000 (without minus)	✓	OK			
•	large_2 shuffled sequence 1, 2,, 100000 (without minus)	√	ОК			
•	large_3 chaotic + many -1, 1, 2, 3 (with minus)	X	TIMEOUT ERROR running time: 0.444 sec., time limit: 0.160 sec.			

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.