Codility_

CodeCheck Report: trainingDP2TTR-72N

Test Name:

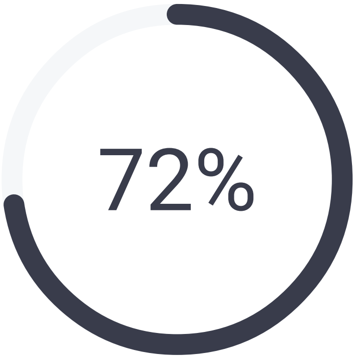
Check out Codility training tasks

Summary Timeline AI Assistant Transcript

Tasks summary

Task		Time spent	Score
FirstUnique C#	⚠	4 min	72%

Total score



Tasks Details

	1. FirstUnique	Task Score	Correctness	Performance
Easy	Find the first unique number in a given sequence.	72%	100%	40%

Task description

A non-empty array A consisting of N integers is given. The *unique number* is the number that occurs exactly once in array A.

For example, the following array A:

```
  A[0] = 4
  A[1] = 10
  A[2] = 5
  A[3] = 4
  A[4] = 2
  A[5] = 10
```

contains two unique numbers (5 and 2).

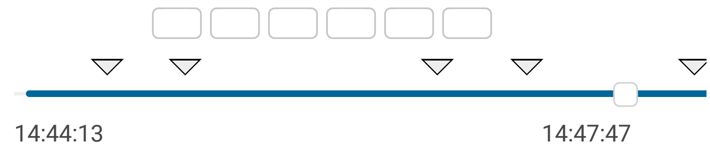
You should find the first unique number in A. In other words, find the unique number with the lowest position in A.

For above example, 5 is in second position (because A[2] = 5) and 2 is in fourth position (because A[4] = 2). So, the first unique number is 5.

Solution

Programming language used:	C#
Total time used:	4 minutes
Effective time used:	4 minutes
Notes:	*not defined yet*

Task timeline



Write a function:

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

that, given a non-empty array A of N integers, returns the first unique number in A. The function should return −1 if there are no unique numbers in A.

For example, given:

```
A[0] = 1
A[1] = 4
A[2] = 3
A[3] = 3
A[4] = 1
A[5] = 2
```

the function should return 4. There are two unique numbers (4 and 2 occur exactly once). The first one is 4 in position 1 and the second one is 2 in position 5. The function should return 4 bacause it is unique number with the lowest position.

Given array A such that:

```
A[0] = 6
A[1] = 4
A[2] = 4
A[3] = 6
```

the function should return −1. There is no unique number in A (4 and 6 occur more than once).

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 [0..1,000,000,000].

Copyright 2009–2024 by Codility Limited. All Rights Reserved. Unauthorized copying, publication or disclosure prohibited.

Code: 14:47:47 UTC, cs, final, score: 72

show code in pop-up

```
using System;
using System.Collections.Generic;

class Solution {
    public int solution(int[] A)
    {
        List<int> uniqueIntegersList = new List<int
        List<int> duplicateIntegersList = new List<
        int returnValue = -1;

        foreach (int number in A)
        {
            if (uniqueIntegersList.Contains(number)
            {
                uniqueIntegersList.Remove(number);
                duplicateIntegersList.Add(number);
            }
            else
            {
                if (!duplicateIntegersList.Contains
                {
                    uniqueIntegersList.Add(number);
                }
            }
        }

        if (uniqueIntegersList.Count > 0)
        {
            returnValue = uniqueIntegersList[0];
        }

        return returnValue;
    }
}
```

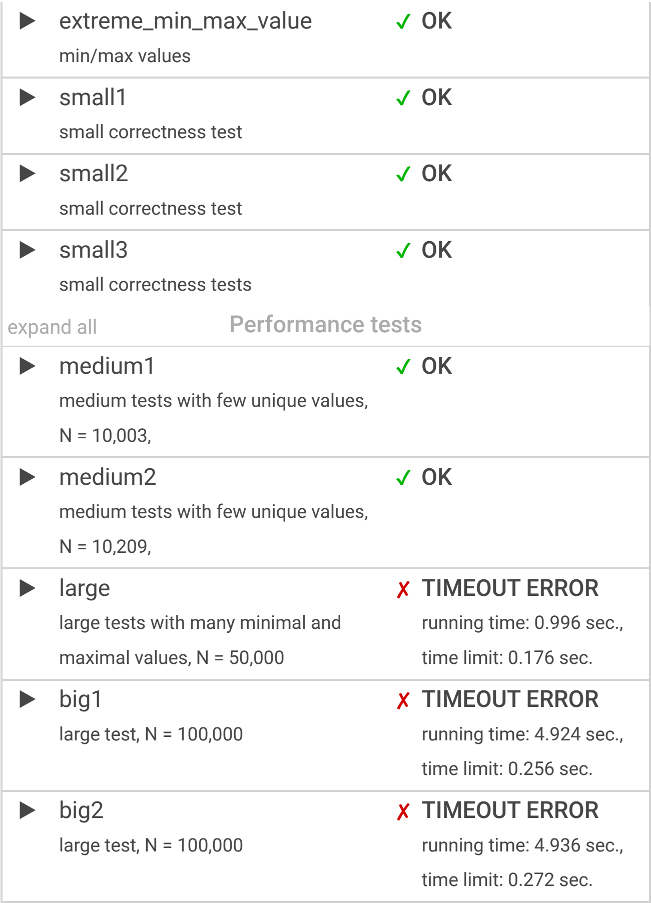
Analysis summary

The following issues have been detected: timeout errors.

Analysis

Detected time complexity: **O(N**2)**

expand all	Example tests	
▶	example0 example	✓ OK
▶	example1 example	✓ OK
▶	example2 example	✓ OK
expand all	Correctness tests	
▶	extreme_single single element	✓ OK
▶	extreme_no_unique no unique value and [1,2,3,4]	✓ OK



▶	extreme_min_max_value min/max values	✓ OK
▶	small1 small correctness test	✓ OK
▶	small2 small correctness test	✓ OK
▶	small3 small correctness tests	✓ OK
expand all	Performance tests	
▶	medium1 medium tests with few unique values, N = 10,003,	✓ OK
▶	medium2 medium tests with few unique values, N = 10,209,	✓ OK
▶	large large tests with many minimal and maximal values, N = 50,000	✗ TIMEOUT ERROR running time: 0.996 sec., time limit: 0.176 sec.
▶	big1 large test, N = 100,000	✗ TIMEOUT ERROR running time: 4.924 sec., time limit: 0.256 sec.
▶	big2 large test, N = 100,000	✗ TIMEOUT ERROR running time: 4.936 sec., time limit: 0.272 sec.