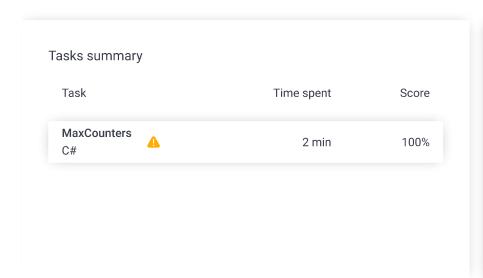
Codility_

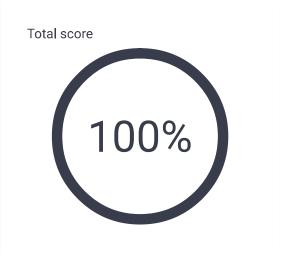
CodeCheck Report: trainingM4G67B-9YH

Test Name:

Summary Timeline

Check out Codility training tasks





Tasks Details

1. MaxCounters

Calculate the values of counters after applying all alternating operations: increase counter by 1; set value of all counters to current maximum.

Task Score

Correctness

Performance

100%

100%

Task description

You are given N counters, initially set to 0, and you have two possible operations on them:

- increase(X) counter X is increased by 1,
- max counter all counters are set to the maximum value of any counter.

A non-empty array A of M integers is given. This array represents consecutive operations:

- if A[K] = X, such that 1 ≤ X ≤ N, then operation K is increase(X),
- if A[K] = N + 1 then operation K is max counter.

For example, given integer N = 5 and array A such that:

Solution

Programming language used: C#

Total time used: 2 minutes

Effective time used: 2 minutes

Notes: not defined yet

Task timeline

```
A[0] = 3
A[1] = 4
A[2] = 4
A[3] = 6
A[4] = 1
A[5] = 4
A[6] = 4
```

the values of the counters after each consecutive operation will be:

```
(0, 0, 1, 0, 0)
(0, 0, 1, 1, 0)
(0, 0, 1, 2, 0)
(2, 2, 2, 2, 2)
(3, 2, 2, 2, 2)
(3, 2, 2, 3, 2)
(3, 2, 2, 4, 2)
```

The goal is to calculate the value of every counter after all operations.

Write a function:

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

that, given an integer N and a non-empty array A consisting of M integers, returns a sequence of integers representing the values of the counters.

Result array should be returned as an array of integers.

For example, given:

A[0] = 3 A[1] = 4 A[2] = 4 A[3] = 6 A[4] = 1 A[5] = 4 A[6] = 4

the function should return [3, 2, 2, 4, 2], as explained above.

Write an efficient algorithm for the following assumptions:

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

Copyright 2009–2022 by Codility Limited. All Rights Reserved. Unauthorized copying, publication or disclosure prohibited.

04:05:13 04:06:37

show code in pop-up

Code: 04:06:36 UTC, cs,

```
final, score: 100
1
     using System;
 2
3
     /* Lesson 4.2 - Max Counter
4
      * Paulo Santos
5
      * 24.Nov.2022
6
7
     class Solution {
         public int[] solution(int N, int[] A) {
8
9
10
11
                       * Check the inputs
                       */
12
13
             if (A == null)
                  throw new ArgumentNullException("A is I
14
15
                      if (N < 0)
                               throw new ArgumentOutOfRang
16
17
             var cnt = new int[N]; // counters
18
             var mct = new int[N]; // indicate the need
19
20
             var max = 0;
                                     // current max counte
                                     // number of times th
21
             var mcc = 0;
22
             var mud = 0;
                                     // max counter for up
              for(var i = 0; i < A.Length; i++) {</pre>
23
24
                   * Check the command to see if it's to
25
                   */
26
27
                  var ord = A[i] - 1;
28
                  if (\emptyset <= ord && ord <= (N - 1)) {
29
                       * Check the need to max the count
30
                       */
31
32
                      if (mct[ord] != mcc) {
33
                          mct[ord] = mcc;
                          cnt[ord] = mud;
34
35
36
                      max = Math.Max(max, ++cnt[ord]);
                  }
37
38
                  else
39
                  {
40
41
                       * Add the count to max the counter
                       */
42
                      mcc++;
43
44
                      mud = max;
45
                  }
46
             }
47
48
               * Adjust the counters that were not maxed
49
50
               */
51
              for (var i = 0; i < cnt.Length; i++)
52
                  if (mct[i] != mcc)
53
                      cnt[i] = mud;
54
55
             return cnt;
56
         }
57
     }
```

Analysis summary

The solution obtained perfect score.

Analysis

Detected time complexity: O(N + M)

ехра	nd all	Example test	S	
•	example example test			OK
expand all Correctness tests				
•	extreme_small all max_counter ope	erations	✓	OK
•	single only one counter		✓	OK
•	small_random1 small random test, 6 operations	o max_counter	✓	OK
•	small_random2 small random test, 1 operations	0 max_counter	✓	OK
expand all Performance tests				
•	medium_random medium random tes operations		✓	OK
•	medium_randon medium random tes operations		√	OK
•	large_random1 large random test, 2 operations	120 max_counter	✓	OK
•	large_random2 large random test, 1 operations	0000 max_counter	✓	OK
•	extreme_large all max_counter ope	rations	✓	OK