Codility_

Candidate Report: trainingYU2QGF-QYA

Check out Codility training tasks

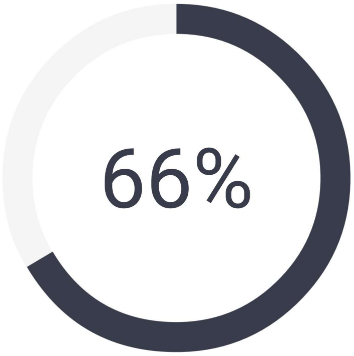
Test Name:

Summary Timeline Feedback

Tasks summary

Task	Time spent	Score
MaxCounters C#	1 min	66%

Total score



Tasks Details

1. MaxCounters

Medium

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
66%	100%	40%

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:

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

Solution

Programming language used:	C#
Total time used:	1 minutes
Effective time used:	1 minutes
Notes:	*not defined yet*

Task timeline

02:34:41 02:35:32

Code: 02:35:32 UTC, cs, final, score: **66**

show code in pop-up

```
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].

```
using System;
using System.Collections.Generic;
using System.Text;
using System.Linq;
// you can also use other imports, for example:
// using System.Collections.Generic;

// you can write to stdout for debugging purposes, e.g.
// Console.WriteLine("this is a debug message");

class Solution {
    private int [] setCountersArray(int N, int value)
        {
            List<int> counters = new List<int>();
            for (int i = 1; i <= N; i++)
            {
                counters.Add(value);
            }
            return counters.ToArray();
        }
        public int[] solution(int N, int[] A)
        {
            var countersArray = setCountersArray(N, 0);
            int upper = A.Length;
            for (int i=0; i < upper; i++)
            {
                if (A[i] == N + 1)
                {
                    countersArray = setCountersArray(N, cou
                }else
                {
                    countersArray[A[i] - 1] += 1;
                }
            }
            return countersArray;
        }
}
```

Analysis summary

The following issues have been detected: timeout errors.

Analysis ?

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

expand all **Example tests**

▶ example ✓ OK
example test

expand all **Correctness tests**

▶ extreme_small ✓ OK
all max_counter operations

▶ single ✓ OK
only one counter

▶ small_random1 ✓ OK
small random test, 6 max_counter operations

▶ small_random2 ✓ OK
small random test, 10 max_counter operations

expand all **Performance tests**

▶	medium_random1 medium random test, 50 max_counter operations	✓ OK
▶	medium_random2 medium random test, 500 max_counter operations	✓ OK
▶	large_random1 large random test, 2120 max_counter operations	✗ TIMEOUT ERROR running time: 1.564 sec., time limit: 0.128 sec.
▶	large_random2 large random test, 10000 max_counter operations	✗ TIMEOUT ERROR Killed. Hard limit reached: 6.000 sec.
▶	extreme_large all max_counter operations	✗ TIMEOUT ERROR Killed. Hard limit reached: 6.000 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.