

Codility

CodeCheck Report: training22AJ2R-H42

Test Name:

[Check out Codility training tasks](#)

Summary

Timeline

Tasks summary

Task		Time spent	Score
MaxCounters Python	⚠	19 min	100%

Total score



Tasks Details

Medium	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%	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

Solution

Programming language used: Python

Total time used: 19 minutes ?

Effective time used: 19 minutes ?

represents consecutive operations:

- if $A[K] = X$, such that $1 \leq X \leq N$, 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
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:

```
def solution(N, 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]`.

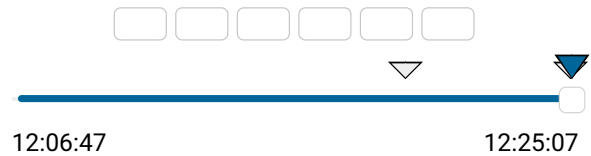
Copyright 2009–2023 by Codility Limited. All Rights Reserved.

Unauthorized copying, publication or disclosure prohibited.

Notes:

not defined yet

Task timeline



Code: 12:25:06 UTC, py, [show code in pop-up](#)
final, score: 100

```
1 # you can write to stdout for debugging
2 # print("this is a debug message")
3
4 def solution(N, A):
5     # Implement your solution here
6     # pass
7     counters = [0] * N
8     maximum_counter = 0
9     last_maximum_counter = 0
10
11     for operation in A:
12         if 1 <= operation <= N:
13             if counters[operation - 1] <
14                 counters[operation - 1]
15                 counters[operation - 1] += 1
16             if counters[operation - 1] >
17                 maximum_counter = counters[operation - 1]
18         elif operation == N + 1:
19             last_maximum_counter = maximum_counter
20
21     for i in range(N):
22         if counters[i] < last_maximum_counter:
23             counters[i] = last_maximum_counter
24
25     return counters
```

Analysis summary

The solution obtained perfect score.

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** ✓ OK
medium random test, 50
max_counter operations

▶ **medium_random2** ✓ OK
medium random test, 500
max_counter operations

▶ **large_random1** ✓ OK
large random test, 2120
max_counter operations

▶ **large_random2** ✓ OK
large random test, 10000
max_counter operations

▶ **extreme_large** ✓ OK
all max_counter operations