Open Feedback Dialog
Open Feedback Dialog

codility

ning center
t Codility training tasks

Congratulations

You have completed a Codility demo.

Tweet this!
I scored 88% in #ruby on @Codility!
https://codility.com/demo/take-sample-test/max_counters/

Sign up for our newsletter!

Like us on Facebook!

Demo ticket

Session

ID: demoTUTKAX-Q7F
Time limit: 120 min.

Status: closed

Created on: 2014-12-14 08:32 UTC
Started on: 2014-12-14 08:32 UTC
Finished on: 2014-12-14 08:55 UTC

Tasks in test	Correctness	Performance	Task score
1 MaxCounters	100%	80%	88%

Test score

88%

88 out of 100 points

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.

score: 88 of 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 zero-indexed 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
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.

Solution

Programming language used: Ruby

Total time used: 24 minutes

Effective time used: 24 minutes

Notes: *not defined yet*

Task timeline

08:32:14 08:55:40

Code: 08:55:40 UTC, rb, final, score: **88.00**

```ruby
def solution(n, a)
  r = Array.new(n, 0)
  max = 0
  a.each do |e|
    i = e - 1
    if e > n
      r = Array.new(n, max)
      #r.each_with_index do |v, k|
        #r[k] = max
      #end
    else
      r[i] += 1
```

Write a function:

```
def solution(n, a)
```

that, given an integer N and a non-empty zero-indexed array A consisting of M integers, returns a sequence of integers representing the values of the counters.
The sequence should be returned as:

- a structure Results (in C), or
- a vector of integers (in C++), or
- a record Results (in Pascal), or
- an array of integers (in any other programming language).

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.
Assume that:

- 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].

Complexity:

- expected worst-case time complexity is O(N+M);
- expected worst-case space complexity is O(N), beyond input storage (not counting the storage required for input arguments).

Elements of input arrays can be modified.


```
13        max = r[i] if max < r[i]
14      end
15    end
16    r
17  end
```

Analysis

Detected time complexity:

O(N + M)

test	time	result
Example tests		
example example test	0.060 s	**OK**
Correctness tests		
extreme_small all max_counter operations	0.056 s	**OK**
single only one counter	0.064 s	**OK**
small_random1 small random test, 6 max_counter operations	0.060 s	**OK**
small_random2 small random test, 10 max_counter operations	0.056 s	**OK**
Performance tests		
medium_random1 medium random test, 50 max_counter operations	0.064 s	**OK**
medium_random2 medium random test, 500 max_counter operations	0.068 s	**OK**
large_random1 large random test, 2120 max_counter operations	0.292 s	**OK**
large_random2 large random test, 10000 max_counter operations	0.892 s	**OK**
extreme_large all max_counter operations	>7.000 s	**TIMEOUT ERROR** running time: >7.00 sec., time limit: 1.52 sec.

Training center