Demo ticket

Session

ID: demo6VRT5R-YVV
Time limit: 120 min.

Status: closed

Created on: 2014-04-25 03:54 UTC
Started on: 2014-04-25 03:54 UTC
Finished on: 2014-04-25 04:02 UTC

Tasks in test	Correctness	Performance	Task score
1 | {} MaxCounters	100%	100%	100%

Test score

100%

100 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: 100 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)
```

Solution

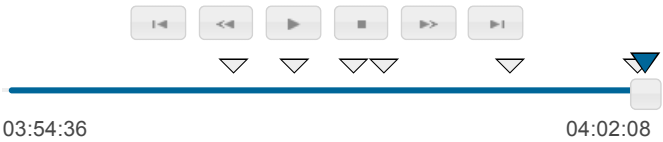
Programming language used: Python

Total time used: 8 minutes

Effective time used: 8 minutes

Notes: correct functionality and scalability

Task timeline



03:54:36 04:02:08

Code: 04:02:08 UTC, py, final, score: **100.00**

```
01. def solution(N, A):
02.     counters = [0] * N
03.     cur_max = 0
04.     max_op = 0
05.
06.     for v in A:
07.         if v <= N:
08.             if max_op > counters[v-1]:
09.                 counters[v-1] = max_op + 1
10.             else:
11.                 counters[v-1] += 1
```

```
(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 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.


```
12.
13.                 if cur_max < counters[v-1]:
14.                     cur_max = counters[v-1]
15.             if v == (N+1):
16.                 max_op = cur_max
17.
18.     for i in xrange(N):
19.         if counters[i] < max_op:
20.             counters[i] = max_op
21.
22.     return counters
```

Analysis

Detected time complexity:
O(N + M)

test	time	result
Example tests		
example example test	0.050 s.	OK
Correctness tests		
extreme_small all max_counter operations	0.050 s.	OK
single only one counter	0.050 s.	OK
small_random1 small random test, 6 max_counter operations	0.050 s.	OK
small_random2 small random test, 10 max_counter operations	0.050 s.	OK
Performance tests		
medium_random1 medium random test, 50 max_counter operations	0.050 s.	OK
medium_random2 medium random test, 500 max_counter operations	0.060 s.	OK
large_random1 large random test, 2120 max_counter operations	0.180 s.	OK
large_random2 large random test, 10000 max_counter operations	0.340 s.	OK
extreme_large all max_counter operations	0.460 s.	OK

Training center