Open Feedback Dialog

codility

Training center

Check out Codility training tasks

Demo ticket

Session

ID: demoC87N2Y-37H
Time limit: 120 min.

Status: closed

Created on: 2014-03-17 19:01 UTC
Started on: 2014-03-17 19:01 UTC
Finished on: 2014-03-17 19:06 UTC

Tasks in test

Task score

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)
(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)
```

Solution

Programming language used: Python

Total time used: 5 minutes

Effective time used: 5 minutes

Notes: correct functionality and scalability

Task timeline

19:01:45 19:06:15

Code: 19:06:15 UTC, py, final, score: **100.00**

```
01. def solution(N, A):
02.     dic = {}
03.     base = 0
04.     max_count = 0
05.     for a in A:
06.         if a == N + 1:
07.             base += max_count
08.             dic = {}
09.             max_count = 0
10.         else:
11.             if a in dic:
12.                 dic[a] += 1
13.             else:
14.                 dic[a] = 1
15.             max_count = max(max_count, dic[a])
16.
17.     ret = [base] * N
18.     for key in dic.keys():
19.         ret[key-1] += dic[key]
20.     return ret
```

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.

Analysis

Detected time complexity:
O(N + M)

test	time	result
example example test	0.050 s.	OK
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
medium_random1 medium random test, 50 max_counter operations	0.050 s.	OK
medium_random2 medium random test, 500 max_counter operations	0.050 s.	OK
large_random1 large random test, 2120 max_counter operations	0.170 s.	OK
large_random2 large random test, 10000 max_counter operations	0.330 s.	OK
extreme_large all max_counter operations	0.450 s.	OK

Training center