Congratulations

You have completed a Codility training test.

Tweet this!
I scored 100% in #java on @Codility!
https://codility.com/demo/take-sample-test/max_counters/

Sign up for our newsletter!

Like us on Facebook!

Training ticket

Session

ID: trainingP9VETG-TEX
Time limit: 120 min.

Status: closed

Created on: 2016-06-22 17:20 UTC
Started on: 2016-06-22 17:20 UTC
Finished on: 2016-06-22 17:21 UTC

Tasks in test	Correctness	Performance	Task score
1 MaxCounters Submitted in: Java	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)
```

Solution

Programming language used: Java

Total time used: 1 minutes

Effective time used: 1 minutes

Notes: *not defined yet*

Task timeline

17:20:57 17:21:44

Code: 17:21:44 UTC, java, final, score: 100

show code in pop-up

```java
import java.util.Optional;
import java.util.stream.IntStream;

class Solution {
    public int[] solution(int N, int[] A) {
        int[] counters = new int[N];
        int maxCounter = -Integer.MIN_VALUE;
    (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 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.

```
        Optional<Integer> updateValue = Optional.empty

        for (int counterId : A) {
            if (counterId > N) {
                updateValue = Optional.of(maxCounter);
            } else {
                updateValue.ifPresent((Integer syncVal
                    if (counters[counterId - 1] < sync
                        counters[counterId - 1] = sync
                    }
                });
                int newCounterValue = ++counters[count
                maxCounter = Math.max(maxCounter, newC
            }
        }

        Integer lastSyncValue = updateValue.orElseGet(
        return IntStream.of(counters).map(v -> Math.ma
    }
}
```

Analysis summary

The solution obtained perfect score.

Analysis

Detected time complexity:
O(N + M)

expand all	Example tests	
▶	example example test	✔ OK

expand all	Correctness tests	
▶	extreme_small all max_counter operations	✔ OK
▶	single only one counter	✔ OK
▶	small_random1 small random test, 6 max_counter operations	✔ OK
▶	small_random2 small random test, 10 max_counter operations	✔ OK

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	✔ OK
▶	large_random2 large random test, 10000 max_counter operations	✔ OK
▶	extreme_large all max_counter operations	✔ OK

Training center