



CodeCheck Report: trainingQR8RAS-DDP

[Check out Codility training tasks](#)

Test Name:

Summary Timeline

Tasks summary

Task	Time spent	Score
MaxCounters Java 8	3 min	66%

Total score

66%

Tasks Details

1. MaxCounters

Calculate the values of counters after applying all alternating operations: increase counter by 1; set value of all counters to current maximum.

Medium

Task Score	Correctness	Performance
66%	100%	40%

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 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

Solution

Programming language used: Java 8

Total time used: 3 minutes ?

Effective time used: 3 minutes ?

Notes: not defined yet

Task timeline

01:02:04 01:05:02

Code: 01:05:01 UTC, java, [show code in pop-up](#)
final, score: 66

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:

```
class Solution { public int[] solution(int N, int[] 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 2009–2022 by Codility Limited. All Rights Reserved. Unauthorized copying, publication or disclosure prohibited.

```
1 import java.util.*;
2 import java.util.Arrays;
3
4 class Solution {
5     // https://github.com/burakozcelik5028/JavaHel
6     public static int[] solution(int N, int[] A) {
7         int[] B = new int[N];
8         Arrays.fill(B, 0, N, 0);
9
10        for(int x= 0; x < A.length; x++) {
11            if(A[x] <= N && A[x] >= 1)
12                B[A[x]-1] = B[A[x]]
13            } else if(A[x] == N+1) {
14                int max = maximum(
15                    Arrays.fill(B, 0,
16
17                )
18
19            }
20        }
21        return B;
22    }
23    public static int maximum(int[] numbers)
24    {
25        int maxSoFar = numbers[0];
26
27        // for each loop
28        for (int num : numbers)
29        {
30            if (num > maxSoFar)
31            {
32                maxSoFar = num;
33            }
34        }
35        return maxSoFar;
36    }
37 }
```

Analysis summary

The following issues have been detected: timeout errors.

Analysis

Detected time complexity: **O(N*M)**

Example tests	
▶ example	✓ OK
example test	
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	
Performance tests	
▶ medium_random1	✓ OK
medium random test, 50 max_counter operations	

Test results - Codility

▶ medium_random2	✓ OK
medium random test, 500 max_counter operations	
▶ large_random1	✗ TIMEOUT ERROR
large random test, 2120 max_counter operations	
running time: 2.124 sec., time limit: 1.648 sec.	
▶ large_random2	✗ TIMEOUT ERROR
large random test, 10000 max_counter operations	
Killed. Hard limit reached: 8.000 sec.	
▶ extreme_large	✗ TIMEOUT ERROR
all max_counter operations	
Killed. Hard limit reached: 11.000 sec.	