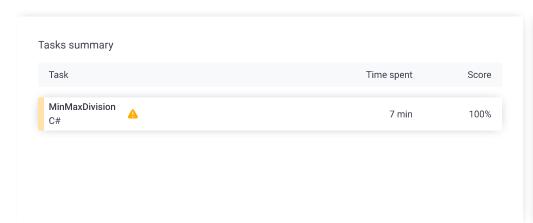
CodeCheck Report: trainingA482UB-VAJ

Test Name:

Summary

Timeline





Check out Codility training tasks

Tasks Details

Medium

1. MinMaxDivision

Divide array A into K blocks and minimize the largest sum of any block.



Task description

You are given integers K, M and a non-empty array A consisting of N integers. Every element of the array is not greater than M.

You should divide this array into K blocks of consecutive elements. The size of the block is any integer between 0 and N. Every element of the array should belong to some block.

The sum of the block from X to Y equals A[X] + A[X + 1] + ... + A[Y]. The sum of empty block equals 0.

The large sum is the maximal sum of any block.

For example, you are given integers K = 3, M = 5 and array A such that:

- A[0] = 2
- A[1] = 1
- A[2] = 5
- A[3] = 1
- A[4] = 2
- A[5] = 2A[6] = 2

The array can be divided, for example, into the following blocks:

- [2, 1, 5, 1, 2, 2, 2], [], [] with a large sum of 15;
- [2], [1, 5, 1, 2], [2, 2] with a large sum of 9;
- [2, 1, 5], [], [1, 2, 2, 2] with a large sum of 8;
- [2, 1], [5, 1], [2, 2, 2] with a large sum of 6.

The goal is to minimize the large sum. In the above example, 6 is the minimal large sum.

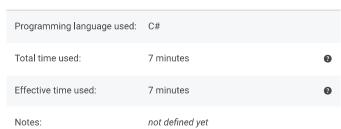
Write a function:

class Solution { public int solution(int K, int M, int[] A); }

that, given integers K, M and a non-empty array A consisting of N integers, returns the minimal large sum.

Solution

Task timeline





```
Code: 05:28:38 UTC, cs, final, score:
                                                      show code in pop-up
100
     using System;
     using System.Linq;
4
      * 14.1 - Min Max Division
      * Paulo Santos
      * 10.Jan.2023
     class Solution {
10
         public int solution(int K, int M, int[] A) {
11
12
             var st = A.Max();
             var ed = A.Sum();
13
             while (st <= ed) {
15
                  var md = (st + ed) / 2;
                  if (VerifyBlocks(A, md) < K)</pre>
17
                      ed = md - 1;
```

For example, given K = 3, M = 5 and array A such that:

A[0] = 2 A[1] = 1 A[2] = 5 A[3] = 1 A[4] = 2 A[5] = 2 A[6] = 2

the function should return 6, as explained above.

Write an efficient algorithm for the following assumptions:

- N and K are integers within the range [1..100,000];
- M is an integer within the range [0..10,000];
- each element of array A is an integer within the range [0..M].

Copyright 2009–2023 by Codility Limited. All Rights Reserved. Unauthorized copying, publication or disclosure prohibited.

```
18
                else
                    st = md + 1;
19
20
21
             return st;
        }
22
23
        private int VerifyBlocks(int[] A, int maxSum) {
24
25
26
            var ct = 0;
27
             for (var i = 0; i < A.Length; i++) {
                if (sm + A[i] > maxSum) {
28
29
                    sm = 0;
30
                    ct += 1;
31
32
                sm += A[i];
33
34
            return ct;
35
        }
36
    }
```

Analysis summary

The solution obtained perfect score.

Analysis

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

	Ja Annaha
and all Examp ► example	ole tests ✓ OK
example test	V OK
and all Correctr	ness tests
extreme_single single elements	√ OK
 extreme_double single and double elements 	√ OK
extreme_min_max maximal / minimal values	√ OK
simple1 simple tests	√ OK
simple2 simple tests	√ OK
tiny_random_ones random values {0, 1}, N = 100	√ OK
and all Perform	ance tests
small_random_ones random values {0, 1}, N = 100	√ OK
medium_zeros many zeros and 99 in the middle, length	✓ OK = 15,000
medium_random random values {1, 100}, N = 20,000	√ OK
large_random random values {0,, MAX_INT}, N = 100,	✓ OK
large_random_ones random values {0, 1}, N = 100,000	√ OK
all_the_same all the same values, N = 100,000	√ OK