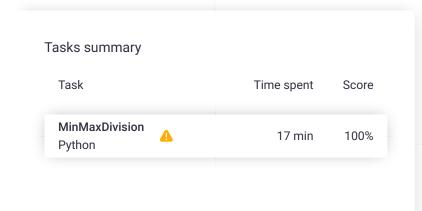
# Codility\_

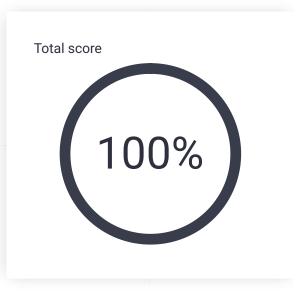
### CodeCheck Report: trainingEERBVX-H8Z

Test Name:

**Timeline** Summary

Check out Codility training tasks





#### **Tasks Details**

#### 1. MinMaxDivision

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

**Task Score** 

100%

Correctness

Performance

100%

#### 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

## Solution

Programming language used: Python

100%

17 minutes Total time used:

Effective time used: 17 minutes

Notes: not defined yet

Task timeline

1 von 3 19.07.23, 10:03 A[1] = 1 A[2] = 5 A[3] = 1 A[4] = 2 A[5] = 2 A[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:

```
def solution(K, M, A)
```

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

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.

07:44:50 08:01:31

```
Code: 08:01:31 UTC, py,
                             show code in pop-up
final, score: 100
     # you can write to stdout for debugging p
     # print("this is a debug message")
     def solution(K, M, A):
         # Implement your solution here
 5
         # pass
6
7
         def check(mid, A, K):
8
             blocks = 1
 9
             block_sum = 0
10
11
             for i in range(len(A)):
12
                  if A[i] > mid:
13
                      return False
                 elif block_sum + A[i] > mid:
14
15
                      block_sum = A[i]
16
                      blocks += 1
                      if blocks > K:
17
18
                          return False
19
20
                      block_sum += A[i]
21
22
             return True
23
24
         start = max(A)
25
         end = sum(A)
26
27
         while start < end:
28
             mid = (start + end) // 2
             if check(mid, A, K):
29
30
                 end = mid
31
             else:
                 start = mid + 1
32
33
34
         return start
```

#### Analysis summary

The solution obtained perfect score.

Analysis

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

expand all	Exampl	e tests	
example example test		<b>∨</b> OK	
expand all Correctne		ess tests	
extreme_s single element	J	<b>∠</b> OK	
extreme_d single and do	ouble ouble elements	<b>∠</b> OK	
extreme_n maximal / mi	_	<b>∠</b> OK	
simple1 simple tests		<b>✓</b> OK	

2 von 3

<b>&gt;</b>	simple2 simple tests tiny_random_ones random values {0, 1}, N = 100		ок	
expand all Performance tests				
<b>&gt;</b>	small_random_ones random values {0, 1}, N = 100	~	ОК	
<b>&gt;</b>	medium_zeros many zeros and 99 in the middle, length = 15,000	~	ОК	
<b>&gt;</b>	medium_random random values {1, 100}, N = 20,000	~	ОК	
<b>&gt;</b>	large_random random values {0,, MAX_INT}, N = 100,000	~	ОК	
<b>&gt;</b>	large_random_ones random values {0, 1}, N = 100,000	~	ОК	
<b>&gt;</b>	all_the_same all the same values, N = 100,000	•	ОК	

3 von 3