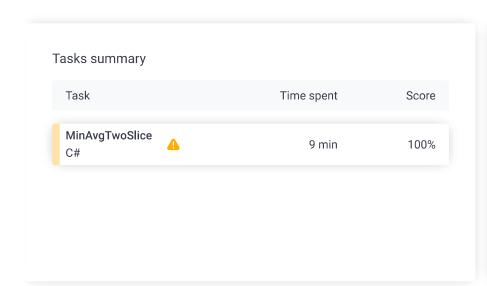
Codility_

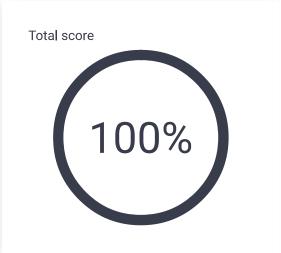
CodeCheck Report: trainingZ63JQB-DQ3

Test Name:

Summary

Timeline





Check out Codility training tasks

Tasks Details

Medium

1. MinAvgTwoSlice

Find the minimal average of any slice containing at least two elements.

Task Score Correctness Performance
100% 100% 100%

Task description

A non-empty array A consisting of N integers is given. A pair of integers (P, Q), such that $0 \le P < Q < N$, is called a *slice* of array A (notice that the slice contains at least two elements). The *average* of a slice (P, Q) is the sum of A[P] + A[P + 1] + ... + A[Q] divided by the length of the slice. To be precise, the average equals (A[P] + A[P + 1] + ... + A[Q]) / (Q - P + 1).

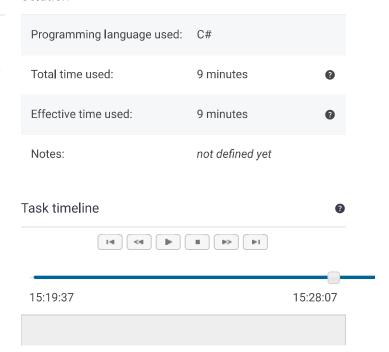
For example, array A such that:

- A[0] = 4
- A[1] = 2
- A[2] = 2
- A[3] = 5
- A[4] = 1
- A[5] = 5
- A[6] = 8

contains the following example slices:

• slice (1, 2), whose average is (2 + 2) / 2 = 2;

Solution



- slice (3, 4), whose average is (5 + 1) / 2 = 3;
- slice (1, 4), whose average is (2 + 2 + 5 + 1) / 4 =

The goal is to find the starting position of a slice whose average is minimal

Write a function:

```
class Solution { public int solution(int[] A); }
```

that, given a non-empty array A consisting of N integers, returns the starting position of the slice with the minimal average. If there is more than one slice with a minimal average, you should return the smallest starting position of such a slice.

For example, given array A such that:

- A[0] = 4
- A[1] = 2
- A[2] = 2
- A[3] = 5
- A[4] = 1
- A[5] = 5
- A[6] = 8

the function should return 1, as explained above.

Write an efficient algorithm for the following assumptions:

- N is an integer within the range [2..100,000];
- each element of array A is an integer within the range [-10,000..10,000].

Copyright 2009–2022 by Codility Limited. All Rights Reserved. Unauthorized copying, publication or disclosure prohibited.

Code: 15:28:06 UTC, cs, final, show code in pop-up score: 100

```
1
     using System;
2
3
      * 5.4 - Min Avg Two Slice
4
5
      * Paulo Santos
6
      * 07.Dec.2022
7
8
      * Thanks to Mr.Duhart
9
                (https://stackoverflow.com/users/4281529/
      */
10
11
     class Solution {
12
         public int solution(int[] A) {
13
14
          * Find prefix sum.
15
16
17
         int N = A.Length;
18
         var ps = new int[N + 1];
19
20
         for (int i = 1; i <= N; i++) {
21
             ps[i] = A[i - 1] + ps[i - 1];
22
23
24
         int lftIdx, minLftIdx;
25
         double avgHere, minAvg, avgOfTwo, avgWithPrev;
26
27
28
          * Initialize variables at the first possible s
29
30
         lftIdx = minLftIdx = 0;
         avgHere = minAvg = (A[0] + A[1]) / 2.0;
31
32
33
34
          * Find min average of every slice that ends at
          * starting at i = 2.
35
36
37
         for (int i = 2; i < N; i ++) {
38
39
40
               * average of A[lftIdx : i]
41
             avgWithPrev = ((double) ps[i + 1] - ps[lftI
42
43
                               (i - lftIdx + 1);
44
45
46
              * average of A[i - 1 : i]
47
             avgOfTwo = (A[i - 1] + A[i]) / 2.0;
48
49
50
              * Find minimum and update lftIdx of slice
51
              * (previous lftIdx or i - 1).
52
53
54
             if (avgOfTwo < avgWithPrev) {</pre>
55
                  avgHere = avgOfTwo;
56
                  lftIdx = i - 1;
57
             }
58
             else
59
                  avgHere = avgWithPrev;
60
61
              * Keep track of minimum so far and its lef
62
              */
63
             if (avgHere < minAvg) {</pre>
64
65
                  minAvg = avgHere;
                  minLftIdx = lftIdx;
66
67
             }
68
         }
69
         return minLftIdx;
70
```

71 72	}	}					

Analysis summary

The solution obtained perfect score.

Analysis

Detected time complexity: O(N)

ехра	nd all Example test	s						
•	example example test	✓	OK					
expand all Correctness tests								
•	double_quadruple two or four elements	✓	OK					
•	simple1 simple test, the best slice has length 3	✓	OK					
•	simple2 simple test, the best slice has length 3	✓	OK					
•	small_random random, length = 100	✓	OK					
•	medium_range increasing, decreasing (legth = ~100) and small functional	✓	OK					
expand all Performance tests								
•	medium_random random, N = ~700	✓	OK					
•	large_ones numbers from -1 to 1, N = \sim 100,000	✓	OK					
>	large_random random, N = ~100,000	√	OK					
•	extreme_values all maximal values, N = ~100,000	√	OK					
•	large_sequence many sequences, N = ~100,000	✓	OK					