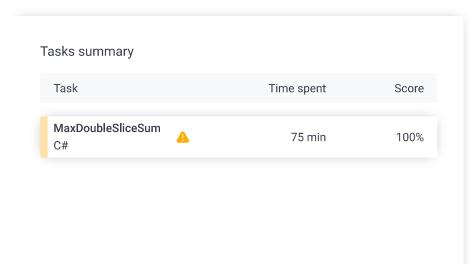
Codility_

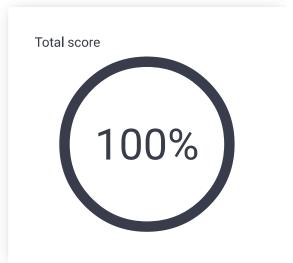
CodeCheck Report: trainingYX63G5-JFS

Test Name:

Timeline Summary







Check out Codility training tasks

Tasks Details

1. MaxDoubleSliceSum Find the maximal sum of

any double slice.

Task Score 100% Correctness 100% Performance

100%

Task description

A non-empty array A consisting of N integers is given.

A triplet (X, Y, Z), such that $0 \le X < Y < Z < N$, is called a *double* slice.

The sum of double slice (X, Y, Z) is the total of A[X + 1] + A[X + 2] +... + A[Y - 1] + A[Y + 1] + A[Y + 2] + ... + A[Z - 1].

For example, array A such that:

A[0] = 3

A[1] = 2

A[2] = 6

A[3] = -1

A[4] = 4

A[5] = 5

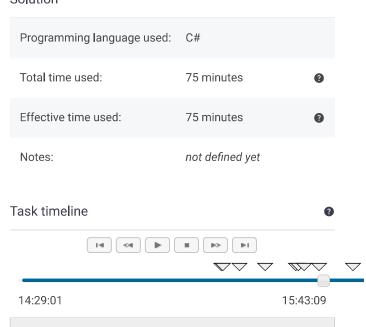
A[6] = -1

A[7] = 2

contains the following example double slices:

• double slice (0, 3, 6), sum is 2 + 6 + 4 + 5 = 17,

Solution



- double slice (0, 3, 7), sum is 2 + 6 + 4 + 5 1 = 16,
- double slice (3, 4, 5), sum is 0.

The goal is to find the maximal sum of any double slice.

Write a function:

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

that, given a non-empty array A consisting of N integers, returns the maximal sum of any double slice.

For example, given:

A[0] = 3

A[1] = 2

A[2] = 6

A[3] = -1

A[4] = 4

A[5] = 5

A[6] = -1

A[7] = 2

the function should return 17, because no double slice of array A has a sum of greater than 17.

Write an efficient algorithm for the following assumptions:

- N is an integer within the range [3..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: 15:43:08 UTC, cs, final,
                                     show code in pop-up
score: 100
1
     using System;
2
3
      * 9.3 - Max Double Slice
4
5
      * Paulo Santos
6
      * 15.Dec.2022
7
8
     class Solution {
9
         public int solution(int[] A) {
10
             var len = A.Length;
11
             var s1Max = new int[len];
12
             var s2Max = new int[len];
13
14
             for(var i = 1; i < A.Length - 1; i++) {</pre>
                  s1Max[i] = Math.Max(s1Max[i - 1] + A[i]
15
16
17
             for(int i = A.Length - 2; i > 0; i--) {
18
                  s2Max[i] = Math.Max(s2Max[i + 1] + A[i]
19
20
21
             int max = 0;
22
23
24
             for(int i = 1; i < A.Length - 1; i++) {</pre>
25
                  max = Math.Max(max, s1Max[i - 1] + s2Ma
26
27
28
             return max;
29
         }
30
```

Analysis summary

The solution obtained perfect score.

Analysis

Detected time complexity: O(N)

expand all	Example tests
example example test	√ OK
expand all	Correctness tests
simple1	√ OK
second simple test	√ OK
simple3 third simple test	√ OK
negative all negative numbers	√ OK
positive all positive numbers	√ OK
•	

	extreme_triplet three elements	✓ OK	
	expand all	Performance tests	5
	small_random1 random, numbers fo length = 70	•	OK
	small_random2 random, numbers free = 300	•	OK
	► medium_range -1000,, 1000	√	ОК
	► large_ones random numbers fro ~100,000	•	OK
	► large_random random, length = ~1	•	ОК
	extreme_maxim all maximal values, I		ОК
	► large_sequence many the same sma length = ~100,000	•	OK