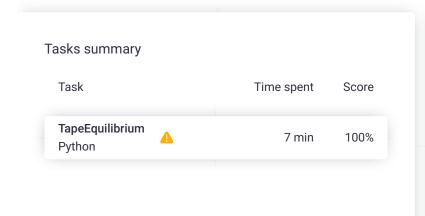
# Codility\_

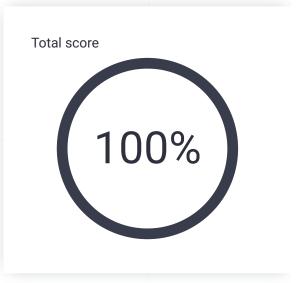
## CodeCheck Report: trainingG9D2FH-952

Test Name:

Summary Timeline

Check out Codility training tasks





## **Tasks Details**

### Task description

A non-empty array A consisting of N integers is given. Array A represents numbers on a tape.

Any integer P, such that 0 < P < N, splits this tape into two non-empty parts: A[0], A[1], ..., A[P - 1] and A[P], A[P + 1], ..., A[N - 1].

The difference between the two parts is the value of: |(A[0] + A[1] + A[P-1]) - (A[P] + A[P+1] + A[N-1])|

A[1] + ... + A[P - 1]) - (A[P] + A[P + 1] + ... + A[N - 1])|

In other words, it is the absolute difference between the sum of the first part and the sum of the second part.

For example, consider array A such that:

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

Solution

Programming language used: Python

Total time used: 7 minutes

Effective time used: 7 minutes

Notes: not defined yet

Task timeline

1 von 3 17.07.23, 13:05

$$A[4] = 3$$

We can split this tape in four places:

- P = 1, difference = |3 10| = 7
- P = 2, difference = |4 9| = 5
- P = 3, difference = |6 7| = 1
- P = 4, difference = |10 3| = 7

#### Write a function:

```
def solution(A)
```

that, given a non-empty array A of N integers, returns the minimal difference that can be achieved.

For example, given:

A[0] = 3

A[1] = 1

A[2] = 2

A[3] = 4

A[4] = 3

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 [-1,000..1,000].

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

```
10:56:52 11:03:37
```

```
Code: 11:03:37 UTC, py,
                             show code in pop-up
final, score: 100
     # you can write to stdout for debugging p
 2
     # print("this is a debug message")
 3
 4
     def solution(A):
 5
         # Implement your solution here
 6
         # pass
 7
         length\_array = len(A)
 8
         total_sum = sum(A)
 9
         left_sum = 0
         minimum = float('inf')
10
11
         for i in range(length_array - 1):
12
13
             left_sum += A[i]
14
             right_sum = total_sum - left_sum
             difference = abs(left_sum - right
15
16
             minimum = min(minimum, difference
17
18
         return minimum
```

#### Analysis summary

The solution obtained perfect score.

#### **Analysis**

## Detected time complexity: O(N)

expand all Example		ts	
<b>&gt;</b>	example example test	~	ОК
expand all Correctness to			s
<b>&gt;</b>	double two elements	<b>~</b>	ОК
<b>&gt;</b>	simple_positive simple test with positive numbers, length = 5	~	ОК
<b>&gt;</b>	simple_negative simple test with negative numbers, length = 5	~	ОК
<b>&gt;</b>	simple_boundary only one element on one of the sides	~	ОК
<b>&gt;</b>	small_random random small, length = 100	<b>V</b>	ОК
<b>&gt;</b>	small_range range sequence, length = ~1,000	<b>V</b>	ОК
<b>&gt;</b>	small elements	~	ОК

2 von 3 17.07.23, 13:05

expa	expand all Performance tests				
•	medium_random1 random medium, numbers from 0 to 100, length = ~10,000	•	OK		
<b>&gt;</b>	medium_random2 random medium, numbers from -1,000 to 50, length = ~10,000	~	OK		
<b>&gt;</b>	large_ones large sequence, numbers from -1 to 1, length = ~100,000	~	OK		
<b>&gt;</b>	large_random random large, length = ~100,000	<b>'</b>	ок		
<b>&gt;</b>	large_sequence large sequence, length = ~100,000	<b>'</b>	ОК		
<b>&gt;</b>	large_extreme large test with maximal and minimal values, length = ~100,000	•	OK		

3 von 3