

Congratulations

You have completed a Codility demo.

Tweet this!

I scored 100% in #scala on @Codility! https://codility.com/demo/take-sample-test/tape_equilibrium/

Sign up for our newsletter!

Like us on Facebook!

Training center

Check out Codility training tasks

Demo ticket

Session

ID: demoBJHGH9-ZBS Time limit: 120 min.

Status: closed

Created on: 2015-07-21 14:43 UTC Started on: 2015-07-21 14:44 UTC Finished on: 2015-07-21 15:04 UTC

Tasks in test

:= TapeEquilibrium

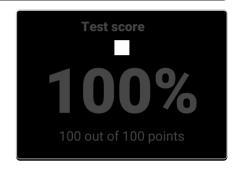
Correctness

100%

Performance 100%

Task score

100%



1. TapeEquilibrium

Minimize the value |(A[0] + ... + A[P-1]) - (A[P] + ... + A[N-1])|.

score: 100 of 100



Task description

A non-empty zero-indexed 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])|

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] = 4A[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:

object Solution { def solution(A: Array[Int]): Int

that, given a non-empty zero-indexed 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] = 4A[4] = 3

the function should return 1, as explained above. Assume that:

• N is an integer within the range [2..100,000];

Solution

Programming language used: Scala

Total time used: 20 minutes

Effective time used: 20 minutes

Notes: not defined vet

Task timeline



15:04:02

 $\overline{\mathcal{M}}$

14.44.13

Code: 15:04:02 UTC, scala, final, score: 100.00 show code in pop-up

```
import scala.collection.
3
    object Solution {
4
     def solution(A: Array[Int]): Int = {
5
       val left = A(0)
6
       val right = A.tail.sum
7
       A.tail.foldLeft((left,right,Set[Int]())) {
8
        9
       }._3.min
10
11
```

Analysis

Detected time complexity:

 each element of array A is an integer within the range [-1,000..1,000].

Complexity:

- expected worst-case time complexity is O(N);
- expected worst-case space complexity is O(N), beyond input storage (not counting the storage required for input arguments).

Elements of input arrays can be modified.

Copyright 2009–2015 by Codility Limited. All Rights Reserved. Unauthorized copying, publication or disclosure prohibited.

expan	d all Example tes	sts	
•	example example test	✓ OK	
expan	d all Correctness t	ests	
•	double two elements	✓ OK	
•	simple_positive simple test with positive numbers, length =	∨ OK 5	
•	simple_negative simple test with negative numbers, length =	∨ OK = 5	
•	small_random random small, length = 100	✓ OK	
•	small_range range sequence, length = ~1,000	∨ OK	
•	small small elements	✓ OK	
expan	d all Performance	tests	
•	medium_random1 random medium, numbers from 0 to 100, length = ~10,000	✓ OK	
•	medium_random2 random medium, numbers from -1,000 to 5 length = ~10,000	∨ OK 0,	
•	large_ones large sequence, numbers from -1 to 1, lengt = ~100,000	✓ OK th	
•	large_random random large, length = ~100,000	✓ OK	

✓ OK

✓ OK

Training center

large_sequence

large_extreme

length = ~100,000

large sequence, length = ~100,000

large test with maximal and minimal values,

© 2009–2015 Codility Ltd., registered in England and Wales (No. 7048726). VAT ID GB981191408. Registered office: 107 Cheapside, London EC2V 6DN