Codility_

CodeCheck Report: trainingG23FQT-WSD

Test Name:

Check out Codility training tasks

Summary Timeline

Tasks summary

Task	Time spent	Score
StrSymmetryPoint Java 8 ⚠	1 min	100%

Total score

Tasks Details

1. **StrSymmetryPoint**
Find a symmetry point of a string, if any.

Easy

Task Score	Correctness	Performance
100%	100%	100%

Task description

Write a function:

```
class Solution { public int solution(String S); }
```

that, given a string S, returns the index (counting from 0) of a character such that the part of the string to the left of that character is a reversal of the part of the string to its right. The function should return −1 if no such index exists.

Note: reversing an empty string (i.e. a string whose length is zero) gives an empty string.

For example, given a string:

"racecar"

the function should return 3, because the substring to the left of the character "e" at index 3 is "rac", and the one to the right is "car".

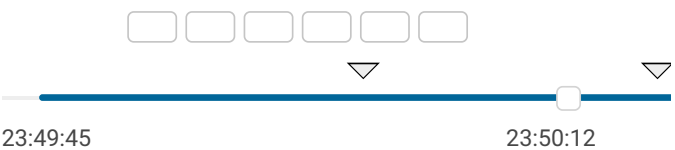
Given a string:

"x"

Solution

Programming language used:	Java 8
Total time used:	1 minutes
Effective time used:	1 minutes
Notes:	*not defined yet*

Task timeline



23:49:45 23:50:12

Code: 23:50:11 UTC, java, final, score: **100** show code in pop-up

the function should return 0, because both substrings are empty.

Write an **efficient** algorithm for the following assumptions:

- the length of string S is within the range [0..2,000,000].

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

```
1  // you can also use imports, for example:
2  // import java.util.*;
3
4  // you can write to stdout for debugging pur
5  // System.out.println("this is a debug messa
6
7  class Solution {
8          public int solution(String S) {
9                  int N = S.length();
10                 if (N % 2 == 0) {
11                         return -1;
12                 }
13
14                 for (int i = 0; i < N /2; i+
15                         if (S.charAt(i) != S
16                                 return -1;
17                         }
18                 }
19
20                 return N / 2;
21         }
22 }
```

Analysis summary

The solution obtained perfect score.

Analysis

Detected time complexity: **O(length(S))**

collapse all	Example tests	
▼ example1 first example		✔ OK
1. 0.004 s	OK	
▼ example2 second example		✔ OK
1. 0.008 s	OK	

collapse all	Correctness tests	
▼ extreme_empty_or_one empty or one character strings		✔ OK
1. 0.004 s	OK	
2. 0.004 s	OK	
▼ symmetric short symmetric strings		✔ OK
1. 0.004 s	OK	
2. 0.008 s	OK	
3.		

0.004 OK
s

▼ even ✔ OK
even length or symmetric strings

1. 0.004 OK
s
2. 0.004 OK
s
3. 0.004 OK
s

▼ three_chars ✔ OK
3 characters (multiple runs)

1. 0.004 OK
s
2. 0.008 OK
s
3. 0.004 OK
s

▼ letters_a ✔ OK
letters 'a' only

1. 0.004 OK
s
2. 0.004 OK
s

▼ alphabet_symmetric ✔ OK
nontrivial symmetry, N = 51

1. 0.004 OK
s

▼ nonsymmetric_inside ✔ OK
mismatch close to the middle, N = 43

1. 0.004 OK
s
2. 0.004 OK
s

▼ nonsymmetric_outside ✔ OK
mismatch close to the ends, N = 43

1. 0.004 OK
s
2. 0.004 OK
s

collapse all Performance tests

▼ large_nonsymmetric ✔ OK
nonsymmetric string, N = 100k+ + [aba]

1. 0.064 OK
s
2. 0.056 OK
s

3.	0.004 s	OK

▼ large_symmetric1 — ✔ OK
symmetric string, N=100k

1.	0.064 s	OK

▼ large_symmetric2 — ✔ OK
symmetric string, N=200k

1.	0.124 s	OK

▼ big_symmetric3 — ✔ OK
symmetric string, N=1M+

1.	0.596 s	OK

▼ big_nonsymmetric — ✔ OK
nonsymmetric string, N = ~1M

1.	0.596 s	OK
2.	0.528 s	OK
3.	0.004 s	OK

▼ extreme_size — ✔ OK
N = ~2M

1.	1.184 s	OK
2.	1.048 s	OK
3.	1.180 s	OK
4.	1.044 s	OK
5.	1.048 s	OK
6.	1.184 s	OK