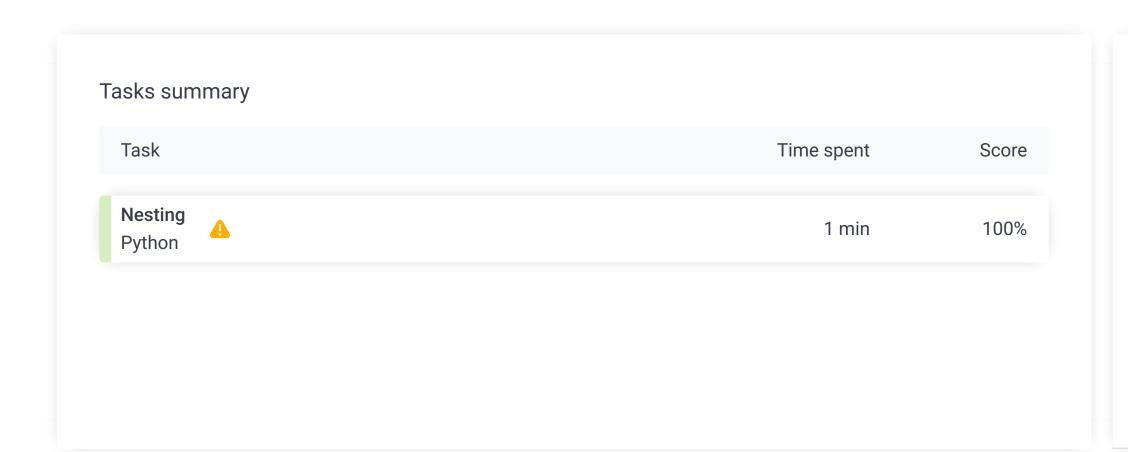
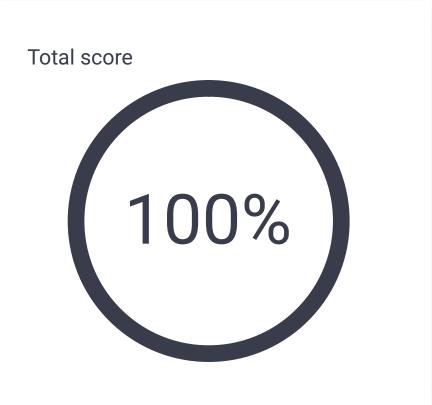
Timeline Summary





Tasks Details



1. Nesting Determine whether a given string of parentheses (single type) is properly nested.



Task description

A string S consisting of N characters is called *properly nested* if:

- S is empty;
- S has the form "(U)" where U is a properly nested string;
- S has the form "VW" where V and W are properly nested strings.

For example, string "(()(())())" is properly nested but string "())" isn't.

Write a function:

def solution(S)

that, given a string S consisting of N characters, returns 1 if string S is properly nested and 0 otherwise.

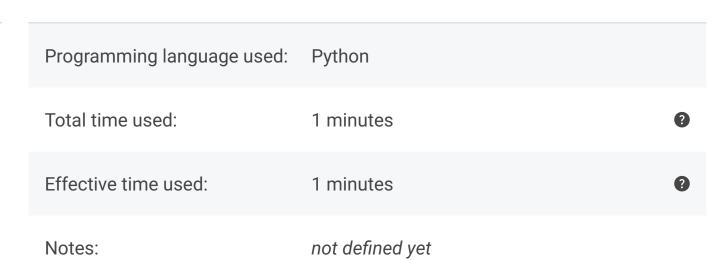
For example, given S = "(()(())())", the function should return 1 and given S = "())", the function should return 0, as explained above.

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

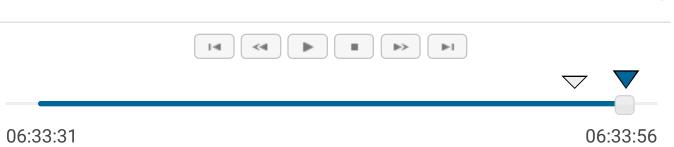
- N is an integer within the range [0..1,000,000];
- string S consists only of the characters "(" and/or ")".

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

Solution



Task timeline



Code: 06:33:55 UTC, py, final, score: show code in pop-up 100		
	1 2	<pre># you can write to stdout for debugging purposes, e.g. # print("this is a debug message")</pre>
	3	<pre># print("this is a debug message")</pre>
	4	<pre>def solution(S):</pre>
	5	# write your code in Python 3.6
	6	if not S:
	7	return 1
	8	if len(S) == 1:
	9	return 0
	10	if len(S) == 2:
	11	if S == '()':
	12	return 1
	13	else:
	14 15	return 0 if S[0] == ')' and S[-1] == '(':
	16	return 0
	17	recurii v
	18	stack_ = []
	19	for c in S:
	20	<pre>if not stack_:</pre>
	21	stack_ += c
	22	else:
	23	if stack_[-1] == '(':
	24	if c == '(':
	25	stack_ += c
	26 27	else:
	28	<pre>stackpop() else:</pre>
	29	if c == ')':
	30	stack_ += c
	31	else:
	32	stack_ += c
	33	
	34	if stack_:
	35	return 0
	36	else:

Analysis summary

The solution obtained perfect score.

return 1

Analysis

37

