

Congratulations

You have completed a Codility demo.

Tweet this!

I scored 100% in #python on @Codility!
<https://codility.com/demo/take-sample-test/nesting/>

Sign up for our newsletter!

Like us on Facebook!

Demo ticket

Session

ID: demoDBE6SR-DCE
 Time limit: 120 min.

Status: closed

Created on: 2014-08-27 03:06 UTC
 Started on: 2014-08-27 03:06 UTC
 Finished on: 2014-08-27 03:15 UTC

Tasks in test

1 | Nesting

Correctness

100%

Performance

100%

Task score

100%

Test score

100%

100 out of 100 points

EASY

1. Nesting

Determine whether given string of parentheses is properly nested.

score: 100 of 100



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. Assume that:

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

Complexity:

- expected worst-case time complexity is $O(N)$;
- expected worst-case space complexity is $O(1)$ (not counting the storage required for input arguments).

Solution

Programming language used: Python

Total time used: 10 minutes

Effective time used: 10 minutes

Notes: correct functionality and scalability

Task timeline



03:06:05

03:15:46

Code: 03:15:46 UTC, py, final, score: 100.00

```
1 def solution(S):
2     left = 0
3     right = 0
4
5     for v in S:
6         if v == '(':
7             left += 1
8         elif v == ')':
9             right += 1
```

```
10         if left < right:
11             return 0
12
13     if left == right:
14         return 1
15
16     return 0
```

Analysis



Detected time complexity:

$O(N)$

test	time	result
Example tests		
example1 example test	0.056 s	OK
example2 example test2	0.056 s	OK
Correctness tests		
negative_match invalid structure, but the number of parentheses matches	0.064 s	OK
empty empty string	0.048 s	OK
simple_grouped simple grouped positive and negative test, length=22	0.056 s	OK
Performance tests		
large1 simple large positive test, 10K ('s followed by 10K 's +) (0.056 s	OK
large2 simple large negative test, 10K+1 ('s followed by 10K 's +) (+ ()	0.048 s	OK
large_full_ternary_tree tree of the form T=(TTT) and depth 11, length=177K+	0.064 s	OK
multiple_full_binary_trees sequence of full trees of the form T=(TT), depths [1..10..1], with/without unmatched ')' at the end, length=49K+	0.048 s	OK
broad_tree_with_deep_paths string of the form (TTT...T) of 300 T's, each T being '(((...)))' nested 200-fold, length=120K+	0.048 s	OK

Training center