Open Feedback Dialog

codility

Training center

Check out Codility training tasks

Demo ticket

Session

ID: demo9SXRHA-KYE
Time limit: 120 min.

Status: closed

Created on: 2014-03-17 19:48 UTC
Started on: 2014-03-17 19:48 UTC
Finished on: 2014-03-17 19:56 UTC

Tasks in test

Task score

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:

```
int solution(const string &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: C++

Total time used: 9 minutes ?

Effective time used: 9 minutes ?

Notes: correct functionality and scalability

Task timeline ?

Code: 19:56:27 UTC, cpp, final, score: **100.00**

```
// you can also use includes, for example:
// #include <algorithm>
int solution(const string &S) {
    long long open_count = 0;

    for (int i = 0; i < (int)S.size(); i++) {
        char ch = S[i];

        if (ch == '(')
            ++open_count;
        else {
            // no more opening p.
            if (open_count == 0)
                return 0;
            else
                --open_count;
        }
    }
    if (open_count > 0)
        return 0;
    else
        return 1;
}
```

Analysis

Detected time complexity:

O(N)

test	time	result
example1 example test	0.020 s.	**OK**
example2 example test2	0.020 s.	**OK**
negative_match invalid structure, but the number of parentheses matches	0.020 s.	**OK**
empty empty string	0.020 s.	**OK**
simple_grouped simple grouped positive and negative test, length=22	0.020 s.	**OK**
large1 simple large positive test, 10K ('s followed by 10K)'s +)(	0.020 s.	**OK**
large2 simple large negative test, 10K+1 ('s followed by 10K)'s +)(+ ()	0.020 s.	**OK**
large_full_ternary_tree tree of the form T=(TTT) and depth 11, length=177K+	0.020 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+	0.020 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+	0.020 s.	**OK**

Training center