1/13/14 Codility

Get account with free plan

See how Codility works from recruiter's point of view.

closed

## Demo ticket

ID: demoF9Z49T-KF5 Time limit: 120 min.

Status: closed

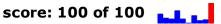
Started on: 2014-01-12 15:19 UTC

Score:

of 100

## 😭 1. Brackets

Determine whether a given string of parentheses is properly nested.



#### **Task description**

A string S consisting of N characters is considered to be properly nested if any of the following conditions is true:

- S has the form "(U)" or "[U]" or "{U}" where U is a properly nested string;
- S has the form " $\mbox{\tt VW}$  where V and W are properly nested strings.

For example, the string " $\{[()()]\}$ " is properly nested but "([)()]" is not.

Write a function:

class Solution { public int solution(String

that, given a string S consisting of N characters, returns 1 if S

explained above.

Assume that:

- N is an integer within the range [0..200,000];
- string S consists only of the following characters: "(", "{", "[", "]", "}" and/or ")".

### Complexity:

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

Copyright 2009–2014 by Codility Limited. All Rights Reserved. Unauthorized copying, publication or disclosure prohibited.

#### Solution

Programming language used: C#

Total time used: 1 minutes

Effective time used: 1 minutes

Notes: correct functionality and scalability

Task timeline



Code: 15:20:31 UTC, cs, final, score: 100.00

```
01.
     using System;
02.
     using System.Collections.Generic;
03.
04.
     class Solution {
05.
            public int solution(string S)
06.
                // write your code in C# with
07.
                    .NET 2.0
08.
                if (S.Length > 200000) throw
                   new
                   ArgumentOutOfRangeException();
                if (String.IsNullOrEmpty(S))
09.
                   return 1;
10.
                var stack = new Stack<char>
                   (S.Length);
11.
                foreach (var value in S)
12.
                {
13.
                    switch (value)
14.
                    {
15.
                        case
                        case '[':
16.
17.
                        case '(
                            stack.Push(value);
18.
19.
                            break;
20.
                        case '}':
21.
                            if(stack.Count == 0
                               || stack.Pop() !=
                                {')
22.
                               return 0;
                           break;
23.
24.
                        case ']':
25.
                            if (stack.Count == 0
                               || stack.Pop() !=
'[')
                               return 0;
26.
27.
                            break;
                        case ')':
28.
```

Codility 1/13/14

```
if (stack.Count == 0
    || stack.Pop() !=
    '(')
    return 0;
29.
30.
31.
                                      break;
32.
                                 default:
33.
                                      return 0;
34.
                            }
35.
                       }
36.
                      if (stack.Count != 0)
    return 0;
37.
38.
39.
40.
                       return 1;
41.
                 }
42. }
```

### **Analysis**

# Detected time complexity:

# O(N)

test	time	result
example1 example test 1	0.080 s.	ок
example2 example test 2	0.080 s.	ок
negative_match invalid structures	0.080 s.	ок
empty empty string	0.080 s.	ок
simple_grouped simple grouped positive and negative test, length=22	0.080 s.	ок
large 1		

Get acco

large1 simple large positive test, 100K ('s followed by 100K )'s + )(	0.080 s.	ок
large2 simple large negative test, $10K+1$ ('s followed by $10K$ )'s + )( + ()	0.080 s.	ок
large_full_ternary_tree tree of the form T=(TTT) and depth 11, length=177K+	0.090 s.	ок
multiple_full_binary_trees sequence of full trees of the form T= (TT), depths [1101], with/without some brackets at the end, length=49K+	0.080 s.	ок
broad_tree_with_deep_paths string of the form [TTTT] of 300 T's, each T being '{{{}}}' nested 200- fold, length=120K+	0.080 s.	ок