

# Valid Parenthesis String

Solution

Given a string containing only three types of characters: '(', ')' and '\*', write a function to check whether this string is valid. We define the validity of a string by these rules:

- Any left parenthesis '(' must have a corresponding right parenthesis ')'.
- Any right parenthesis ')' must have a corresponding left parenthesis '('.
- Left parenthesis '(' must go before the corresponding right parenthesis ')'.
- '\*' could be treated as a single right parenthesis ')' or a single left parenthesis '(' or an empty string.
- An empty string is also valid.

Example 1:

Input: "()"
Output: True

Example 2:

Input: "(\*)"
Output: True

Example 3:

Input: "(\*))"
Output: True

Note:

- The string size will be in the range [1, 100].

Java

```
1 class Solution {
2     public boolean checkValidString(String s) {
3
4         int diff = 0;
5         // left to right, * to open
6         for(char c : s.toCharArray()) {
7             if(c == '(' || c == '*') {
8                 diff++;
9             }
10            else {
11                diff--;
12            }
13            if(diff < 0) return false;
14        }
15
16        if (diff == 0) return true;
17
18        diff = 0;
19        // right to left, * to close
20        for(int i = s.length() - 1; i >= 0; i--) {
21            if(s.charAt(i) == ')' || s.charAt(i) == '*') {
22                diff++;
23            }
24            else {
25                diff--;
26            }
27            if(diff < 0) return false;
28        }
29
30        return true;
31    }
32 }
```

☐ Custom Testcase ( [Contribute](#) )

? Run Code Submit