

678. Valid Parenthesis String

Medium👍 2007💬 54❤️ Add to List🔗 Share

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].

JavaAutocomplete

```
1 class Solution {
2     public boolean checkValidString(String s) {
3         return checkValidString(s, 0, 0, 0);
4     }
5
6     public static boolean checkValidString(String s, int i, int open, int close) {
7         if(s.length() == i) return open == close;
8         if(close > open) return false;
9         switch(s.charAt(i)) {
10             case '(':
11                 return checkValidString(s, i+1, open + 1, close);
12             case ')':
13                 return checkValidString(s, i+1, open, close + 1);
14             case '*':
15                 return checkValidString(s, i+1, open, close) ||
16                     checkValidString(s, i+1, open + 1, close) ||
17                     checkValidString(s, i+1, open, close + 1);
18             }
19         return false;
20     }
21 }
```

TestcaseRun Code ResultDebugger🔒

AcceptedRuntime: 0 ms?

Your input

"()"

Output

true

☐Diff

Expected

true