

# 最长有效括号长度

## 32. 最长有效括号

难度 **困难** 946 喜欢 收藏 评论 举报

给定一个只包含 '(' 和 ')' 的字符串，找出最长的包含有效括号的子串的长度。

示例 1:

输入: "(()"

输出: 2

解释: 最长有效括号子串为 "()"

示例 2:

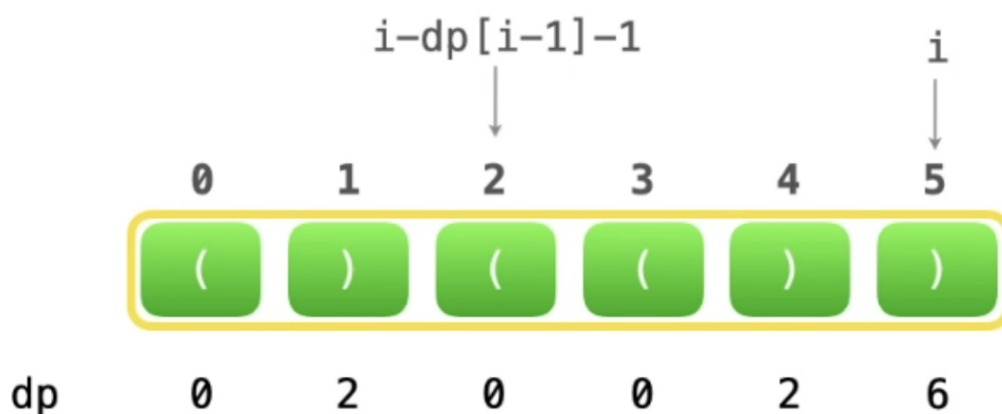
输入: "())(())"

输出: 4

解释: 最长有效括号子串为 "(())"

思路: 重复性,

1 动态规划:



设置dp, 表示i位能够成的有效括号的最大长度

if  $s[i] == \text{'('}$ ; 肯定不能构成有效括号, 为0

那么只考虑 $s[i] == \text{'}'$ 的情况

$dp[i-1]$ 是i-1位能够成有效括号的最大长度,

那么  $i - dp[i-1] - 1$  位如果是 $\text{'('}$ ; 则, 第i位也是参与构成有效括号

注意  $i - dp[i-1] - 1 \geq 0$ , 否则溢出:

状态转移方程:

$dp[i] = 2(\text{左括号和右括号}) + dp[i-1]$  (中间的有效括号长度)  $+ dp[i - dp[i-1] - 2]$  (左括号前的有效括号长度)

```
1 def longestValidParentheses(self, s: str) -> int:
2     size = len(s)
3     if size < 2:
4         return 0
5     dp = [0] * size
6     for i in range(size):
7         if s[i] == '(' and i - dp[i-1] - 1 >= 0 and s[i - dp[i-1] - 1] == ')':
8             dp[i] = dp[i-1] + dp[i - dp[i-1] - 2] + 2
9     return max(dp)
```

2 栈: