

Longest Valid Parentheses

Question: Given a string containing just the characters '(' and ')', find the length of the longest valid (well-formed) parentheses substring.

For "()", the longest valid parentheses substring is "()", which has length = 2.

Another example is "()()()", where the longest valid parentheses substring is "()()", which has length = 4.

Solutions:

class Solution:

@param s, a string

@return an integer

def longestValidParentheses(self, s):

maxlen, stack, last = 0, [], -1

for i in range(len(s)):

if s[i]=='(':

stack.append(i) # push the INDEX into the stack!!!!

else:

if stack == []:

last = i

else:

stack.pop()

if stack == []:

maxlen = max(maxlen, i-last)

else:

maxlen = max(maxlen, i-stack[len(stack)-1])

return maxlen

Solution().longestValidParentheses("()()()")