Problem # 1249 : Minimum Remove to Make Valid Parentheses (Medium)

---------------------------------------------------------------------------------------------------

<https://leetcode.com/problems/minimum-remove-to-make-valid-parentheses/>

My Solution:

Runtime beats 94.25%

1. Since string does not support assignment, convert s to a list.

2. Let n be length of s.

3. Initialize stack to an empty list.

4. Iterate through the list s.

If the character is left parenthesis, then append the index of the character to the stack.

If the character is right parenthesis, then check if the stack is not empty. If it is not empty, pop the stack (i.e. the right parenthesis is balanced by the left parenthesis in the stack). However, if the stack is empty, then make the character to be an empty string.

5. After iterating through the array, if the stack is empty, then get the index numbers from the stack and make the corresponding character in s to be an empty string.

6. Finally join the characters in the list to form a valid string.

-----------------------------------------------

class Solution:

def minRemoveToMakeValid(self, s: str) -> str:

n = len(s)

stack = []

s = list(s)

for i in range(n):

if s[i] == "(":

stack.append(i)

elif s[i] == ')':

if stack:

stack.pop()

else:

s[i] = ''

while stack:

s[stack.pop()] = ''

return ''.join(s)