



链滴

每日 LeetCode] 1021. Remove Outermost Parentheses

作者: [Hanseltu](#)

原文链接: <https://ld246.com/article/1556718708958>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

Description:

A valid parentheses string is either empty `""`, `"(" + A + ")"`, or `A + B`, where `A` and `B` are valid parentheses strings, and `+` represents string concatenation. For example, `"", "()", "(())", and "(())()"` are all valid parentheses strings.

A valid parentheses string `S` is **primitive** if it is nonempty, and there does not exist a way to split it into `S = A+B`, with `A` and `B` nonempty valid parentheses strings.

Given a valid parentheses string `S`, consider its primitive decomposition: `S = P_1 + P_2 + ... + P_k`, where `P_i` are primitive valid parentheses strings.

Return `S` after removing the outermost parentheses of every primitive string in the primitive decomposition of `S`.

Example 1:

Input: `"(())(())"`

Output: `"()()"`

Explanation:

The input string is `"(())(())"`, with primitive decomposition `"(())" + "(())"`.

After removing outer parentheses of each part, this is `"()" + "()" = "()()"`.

Example 2:

Input: `"(())(())(())(())"`

Output: `"()()()()"`

Explanation:

The input string is `"(())(())(())(())"`, with primitive decomposition `"(())" + "(())" + "(())" + "(())"`.

After removing outer parentheses of each part, this is `"()" + "()" + "()" + "()" = "()()()()"`.

Example 3:

Input: `"()"`

Output: `""`

Explanation:

The input string is `"()"`, with primitive decomposition `"()" + "()"`.

After removing outer parentheses of each part, this is `"" + "" = ""`.

Note:

1. `S.length <= 10000`
2. `S[i]` is `"(" or ")"`
3. `S` is a valid parentheses string

思路：本题要求去掉字符串中外层的括号。可以使用栈结构，依次进栈，遇到第一个 `(` 就进栈，第二个 `(` 就加入到返回的字符串中，遇到 `)` 时判断此时栈中的元素个数，如果个数为1则将已有的栈元素弹出，果个数为2则加入返回字符串后弹出。

C++代码

```
class Solution {
public:
    string removeOuterParentheses(string S) {
        string ret = "";
        stack<char> st;
        int size = S.size();
        for (int i=0; i<size; i++){
            if (S[i] == '('){
                if (st.size() > 0)
                    ret += '(';
                st.push('(');
            }
            else{
                if (st.size() > 1)
                    ret += ')';
                st.pop();
            }
        }
        return ret;
    }
};
```

运行时间: 8ms

运行内存: 9.2M