# 题目

给你一个由 '('、')' 和小写字母组成的字符串 s。

你需要从字符串中删除最少数目的 '(' 或者 ')' （可以删除任意位置的括号)，使得剩下的「括号字符串」有效。

请返回任意一个合法字符串。

有效「括号字符串」应当符合以下任意一条要求：

1、空字符串或只包含小写字母的字符串

2、可以被写作 AB（A 连接 B）的字符串，其中 A 和 B 都是有效「括号字符串」

3、可以被写作 (A) 的字符串，其中 A 是一个有效的「括号字符串」

示例 1：

输入：s = "lee(t(c)o)de)"

输出："lee(t(c)o)de"

解释："lee(t(co)de)" , "lee(t(c)ode)" 也是一个可行答案。

示例 2：

输入：s = "a)b(c)d"

输出："ab(c)d"

示例 3：

输入：s = "))(("

输出：""

解释：空字符串也是有效的

提示：

1 <= s.length <= 105

s[i] 可能是 '('、')' 或英文小写字母

# 分析

## 方法一：单调栈

思路：

这个问题可以使用栈来解决。具体的步骤如下：

1、遍历字符串，使用一个栈来记录遇到的左括号 '(' 的索引位置。

2、如果遇到右括号 ')'，首先检查栈是否为空，如果不为空且栈顶元素是左括号 '('，则将栈顶元素出栈，表示匹配成功；否则，将当前右括号的索引位置加入一个要删除的索引集合中。

3、遍历结束后，如果栈中还有左括号未匹配，也将它们的索引加入要删除的索引集合中。

4、遍历字符串，将不在要删除的索引集合中的字符拼接起来，就是最终的结果。

代码：

class Solution {

public:

string minRemoveToMakeValid(string s) {

stack<int> st;

unordered\_set<int> to\_remove;

for (int i = 0; i < s.length(); ++i) {

if (s[i] == '(') {

st.push(i);

} else if (s[i] == ')') {

if (!st.empty()) {

st.pop();

} else {

to\_remove.insert(i);

}

}

}

while (!st.empty()) {

to\_remove.insert(st.top());

st.pop();

}

string result = "";

for (int i = 0; i < s.length(); ++i) {

if (to\_remove.find(i) == to\_remove.end()) {

result += s[i];

}

}

return result;

}

};