Problem 1249: Minimum Remove to Make Valid Parentheses

Problem Information

Difficulty: Medium
Acceptance Rate: 71.12%
Paid Only: No
Tags: String, Stack

Problem Description

Given a string s of `'('` , `')'` and lowercase English characters.

Your task is to remove the minimum number of parentheses (`'('` or `')'`, in any positions) so that the resulting _parentheses string_ is valid and return **any** valid string.

Formally, a _parentheses string_ is valid if and only if:

* It is the empty string, contains only lowercase characters, or * It can be written as `AB` (`A` concatenated with `B`), where `A` and `B` are valid strings, or * It can be written as `(A)`, where `A` is a valid string.

Example 1:

Input: s = "lee(t(c)o)de)" **Output:** "lee(t(c)o)de" **Explanation:** "lee(t(co)de)" , "lee(t(c)ode)" would also be accepted.

Example 2:

Input: s = "a)b(c)d" **Output:** "ab(c)d"

Example 3:

Input: s = "))((" **Output:** "" **Explanation:** An empty string is also valid.

Constraints:

* `1 <= s.length <= 105` * `s[i]` is either `'(' , `')'`, or lowercase English letter.

Code Snippets

C++:

```
class Solution {
public:
string minRemoveToMakeValid(string s) {

}
};
```

Java:

```
class Solution {
public String minRemoveToMakeValid(String s) {

}
}
```

Python3:

```
class Solution:
def minRemoveToMakeValid(self, s: str) -> str:
```