Problem 921: Minimum Add to Make Parentheses Valid

Problem Information

Difficulty: Medium
Acceptance Rate: 74.52%
Paid Only: No
Tags: String, Stack, Greedy

Problem Description

A parentheses string is valid if and only if:

* It is the empty string, * 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.

You are given a parentheses string `s`. In one move, you can insert a parenthesis at any position of the string.

* For example, if `s = "()))"`, you can insert an opening parenthesis to be `"(**(**)))"` or a closing parenthesis to be `"())**)**)"`.

Return _the minimum number of moves required to make_`s` _valid_.

Example 1:

Input: s = "())" **Output:** 1

Example 2:

Input: s = "(((" **Output:** 3

Constraints:

* `1 <= s.length <= 1000` * `s[i]` is either `'('` or `')'`.

Code Snippets

C++:

```
class Solution {
public:
int minAddToMakeValid(string s) {

}
};
```

Java:

```
class Solution {
public int minAddToMakeValid(String s) {

}
}
```

Python3:

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