Problem 1614: Maximum Nesting Depth of the Parentheses

Problem Information

Difficulty: Easy
Acceptance Rate: 84.60%
Paid Only: No
Tags: String, Stack

Problem Description

Given a **valid parentheses string** `s`, return the **nesting depth** of __`s`. The nesting depth is the **maximum** number of nested parentheses.

Example 1:

Input: s = "(1+(2*3)+((8)/4))+1"

Output: 3

Explanation:

Digit 8 is inside of 3 nested parentheses in the string.

Example 2:

Input: s = "(1)+((2))+(((3)))"

Output: 3

Explanation:

Digit 3 is inside of 3 nested parentheses in the string.

Example 3:

Input: s = "()(())((()()))"

Output: 3

Constraints:

* `1 <= s.length <= 100` * `s` consists of digits `0-9` and characters `'+'`, `'-'`, `'*'`, `'/'`, `'('`, and `')'`. * It is guaranteed that parentheses expression `s` is a VPS.

Code Snippets

C++:

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

}
};
```

Java:

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

}
}
```

Python3:

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