[LeetCode](https://leetcode.com/problems/maximum-nesting-depth-of-the-parentheses/?envType=daily-question&envId=2024-04-04)

<https://github.com/AdityaKonda6/-50DaysOfCoding>

<https://leetcode.com/problems/maximum-nesting-depth-of-the-parentheses/?envType=daily-question&envId=2024-04-04>

<https://www.linkedin.com/in/aditya-adi-konda/>

Day 45 of [#50dayscodingchallenge](https://www.linkedin.com/feed/hashtag/?keywords=50dayscodingchallenge&highlightedUpdateUrns=urn:li:activity:7166316239483461633):  
[#leetcode](https://www.linkedin.com/feed/hashtag/?keywords=leetcode&highlightedUpdateUrns=urn:li:activity:7166316239483461633) [#leetcodechallenge](https://www.linkedin.com/feed/hashtag/?keywords=leetcodechallenge&highlightedUpdateUrns=urn:li:activity:7166316239483461633) [#leetcodestreak](https://www.linkedin.com/feed/hashtag/?keywords=leetcodestreak&highlightedUpdateUrns=urn:li:activity:7166316239483461633) [#leetcode2024](https://www.linkedin.com/feed/hashtag/?keywords=leetcode2024&highlightedUpdateUrns=urn:li:activity:7166316239483461633) [#leetcode50day](https://www.linkedin.com/feed/hashtag/?keywords=leetcode50day&highlightedUpdateUrns=urn:li:activity:7166316239483461633)  
   
Ventured further into my coding journey today, tackling the engaging LeetCode Problem "Successfully solved LeetCode Problem ���"

“1614. Maximum Nesting Depth of the Parentheses.”

   
✨ Task: A string is a valid parentheses string (denoted VPS) if it meets one of the following:

It is an empty string "", or a single character not equal to "(" or ")",

It can be written as AB (A concatenated with B), where A and B are VPS's, or

It can be written as (A), where A is a VPS.

We can similarly define the nesting depth depth(S) of any VPS S as follows:

depth("") = 0

depth(C) = 0,

where C is a string with a single character not equal to "(" or ")".

depth(A + B) = max(depth(A), depth(B)), where A and B are VPS's.

depth("(" + A + ")") = 1 + depth(A), where A is a VPS.

For example, "", "()()", and "()(()())" are VPS's (with nesting depths 0, 1, and 2), and ")(" and "(()" are not VPS's.

Given a VPS represented as string s, return the nesting depth of s.

Examples:

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

Let's Connect:

If you find this problem intriguing or have insights to share, let's connect! I'm passionate about problem-solving, algorithmic thinking, and collaborative learning. Feel free to comment or reach out for engaging discussions and knowledge exchange.Unravel the mystery using your coding skills!

[#CodingChallenge](https://www.linkedin.com/feed/hashtag/?keywords=codingchallenge&highlightedUpdateUrns=urn:li:activity:7166316239483461633) [#Algorithm](https://www.linkedin.com/feed/hashtag/?keywords=algorithm&highlightedUpdateUrns=urn:li:activity:7166316239483461633) [#LinkedInPost](https://www.linkedin.com/feed/hashtag/?keywords=linkedinpost&highlightedUpdateUrns=urn:li:activity:7166316239483461633) #Algorithm #Optimization #DataStructures #CodingChallenge  
  
Excited about the progress and challenges ahead!  
   
Make Sure You Follow My GitHub For Solutions: <https://github.com/AdityaKonda6/-50DaysOfCoding>  
  
  
Happy coding!

**Solution:-**

class Solution {

  public int maxDepth(String s) {

    int ans = 0;

    int opened = 0;

    for (final char c : s.toCharArray())

      if (c == '(')

        ans = Math.max(ans, ++opened);

      else if (c == ')')

        --opened;

    return ans;

  }

}

