[LeetCode](https://leetcode.com/problems/minimum-remove-to-make-valid-parentheses/description/?envType=daily-question&envId=2024-04-06)

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

<https://leetcode.com/problems/minimum-remove-to-make-valid-parentheses/description/?envType=daily-question&envId=2024-04-06>

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

Day 47 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 ���"

“1249. Minimum Remove to Make Valid Parentheses.”

   
✨ Task: 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.

Examples:

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.

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 String minRemoveToMakeValid(String s) {

    Deque<Integer> stack = new ArrayDeque<>();

    StringBuilder sb = new StringBuilder(s);

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

      if (sb.charAt(i) == '(') {

        stack.push(i);

      } else if (sb.charAt(i) == ')') {

        if (stack.isEmpty())

          sb.setCharAt(i, '#');

        else

          stack.pop();

      }

    while (!stack.isEmpty())

      sb.setCharAt(stack.pop(), '#');

    return sb.toString().replaceAll("#", "");

  }

}

