Problem 536: Construct Binary Tree from String

Problem Information

Difficulty: Medium
Acceptance Rate: 58.64%
Paid Only: Yes
Tags: String, Stack, Tree, Depth-First Search, Binary Tree

Problem Description

You need to construct a binary tree from a string consisting of parenthesis and integers.

The whole input represents a binary tree. It contains an integer followed by zero, one or two pairs of parenthesis. The integer represents the root's value and a pair of parenthesis contains a child binary tree with the same structure.

You always start to construct the **left** child node of the parent first if it exists.

Example 1:

Input: s = "4(2(3)(1))(6(5))" **Output:** [4,2,6,3,1,5]

Example 2:

Input: s = "4(2(3)(1))(6(5)(7))" **Output:** [4,2,6,3,1,5,7]

Example 3:

Input: s = "-4(2(3)(1))(6(5)(7))" **Output:** [-4,2,6,3,1,5,7]

Constraints:

* `0 <= s.length <= 3 * 104` * `s` consists of digits, `'('`, `')'`, and `'-'` only. * All numbers in the tree have value **at most** than `230`.

Code Snippets

C++:

```
/**
* Definition for a binary tree node.
* struct TreeNode {
* int val;
* TreeNode *left;
* TreeNode *right;
* TreeNode() : val(0), left(nullptr), right(nullptr) {}
* TreeNode(int x) : val(x), left(nullptr), right(nullptr) {}
* TreeNode(int x, TreeNode *left, TreeNode *right) : val(x), left(left),
right(right) {}
* };
*/
class Solution {
public:
TreeNode* str2tree(string s) {

}
};
```

Java:

```
/**
* Definition for a binary tree node.
* public class TreeNode {
* int val;
* TreeNode left;
* TreeNode right;
* TreeNode() {}
* TreeNode(int val) { this.val = val; }
* TreeNode(int val, TreeNode left, TreeNode right) {
* this.val = val;
* this.left = left;
* this.right = right;
* }
```

```
* }
*/
class Solution {
public TreeNode str2tree(String s) {

}
}
```

Python3:

```
# Definition for a binary tree node.
# class TreeNode:
# def __init__(self, val=0, left=None, right=None):
# self.val = val
# self.left = left
# self.right = right
class Solution:
def str2tree(self, s: str) -> Optional[TreeNode]:
```