Description | Solution | Discuss (77) | Submissions

Java | Autocomplete

1597. Build Binary Expression Tree From Infix Expression

Hard 107 24 Add to List Share

A **binary expression tree** is a kind of binary tree used to represent arithmetic expressions. Each node of a binary expression tree has either zero or two children. Leaf nodes (nodes with 0 children) correspond to operands (numbers), and internal nodes (nodes with 2 children) correspond to the operators `'+'` (addition), `'-'` (subtraction), `'*'` (multiplication), and `'/'` (division).

For each internal node with operator `o`, the **infix expression** that it represents is `(A o B)`, where `A` is the expression the left subtree represents and `B` is the expression the right subtree represents.

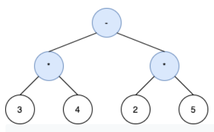
You are given a string `s`, an **infix expression** containing operands, the operators described above, and parentheses `'('` and `')'`.

Return *any valid* ***binary expression tree****, which its in-order traversal reproduces* `s` after omitting the parenthesis from it (see examples below).

Please note that order of operations applies in `s`**.** That is, expressions in parentheses are evaluated first, and multiplication and division happen before addition and subtraction.

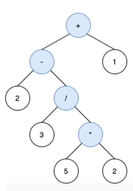
Operands must also appear in the **same order** in both `s` and the in-order traversal of the tree.

Example 1:

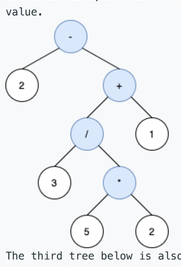



```
Input: s = "3*4-2*5"
Output: [-,*,*,3,4,2,5]
Explanation: The tree above is the only valid tree whose inorder traversal
produces s.
```

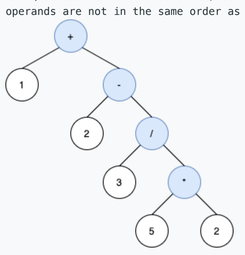
Example 2:




```
Input: s = "2-3/(5*2)+1"
Output: [+,-,1,2,/,null,null,null,null,3,*,null,null,5,2]
Explanation: The inorder traversal of the tree above is 2-3/5*2+1 which is the
same as s without the parenthesis. The tree also produces the correct result and
its operands are in the same order as they appear in s.
The tree below is also a valid binary expression tree with the same inorder
traversal as s, but it not a valid answer because it does not evaluate to the same
value.
```




```
The third tree below is also not valid. Although it produces the same result and
is equivalent to the above trees, its inorder traversal does not produce s and its
operands are not in the same order as s.
```



Example 3:

```
Input: s = "1+2+3+4+5"
Output: [+,+,5,+,4,null,null,+,3,null,null,1,2]
Explanation: The tree [+,+,5,+,+,null,null,1,2,3,4] is also one of many other
valid trees.
```

Constraints:

- `1 <= s.length <= 1000`
- `s` consists of digits and the characters `'+'`, `'-'`, `'*'`, and `'/'`.
- Operands in `s` are **exactly** 1 digit.
- It is guaranteed that `s` is a valid expression.

Accepted 4,763 | Submissions 8,088

Seen this question in a real interview before? Yes No

Companies

0 ~ 6 months 6 months ~ 1 year 1 year ~ 2 years

Amazon | 3

Related Topics

String Stack Tree Binary Tree

Similar Questions

Basic Calculator III Hard

Check If Two Expression Trees are Equivalent Medium

Hide Hint 1

Convert infix expression to postfix expression.

Hide Hint 2

Build an expression tree from the postfix expression.

```java
/**
 * Definition for a binary tree node.
 * class Node {
 *     char val;
 *     Node left;
 *     Node right;
 *     Node() {this.val = ' ';}
 *     Node(char val) { this.val = val; }
 *     Node(char val, Node left, Node right) {
 *         this.val = val;
 *         this.left = left;
 *         this.right = right;
 *     }
 * }
 */
class Solution {
    public Node expTree(String s) {

    }
}
```