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:**

**Input:** "4(2(3)(1))(6(5))"

**Output:** return the tree root node representing the following tree:

4

/ \

2 6

/ \ /

3 1 5

**Note:**

1. There will only be '(', ')', '-' and '0' ~ '9' in the input string.
2. An empty tree is represented by "" instead of "()".