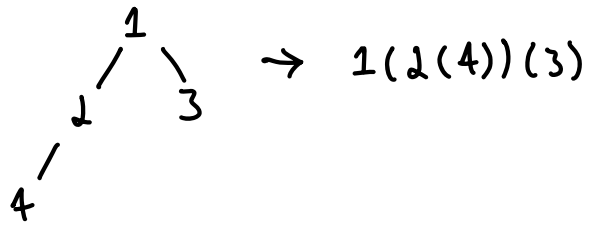
- Given root of binary tree, construct a string of parens & integers in a pre-order traversal way and return it.
- Omit paren pairs that do not affect the 1-1 mapping between string and original tree.



→ 1(2(4))(3)

```
tree2str(tree* root) → str
    if root is null => {}
    result: str = {}root->value; helper(root, result);
    return result
helper(tree* node, result: str) → void
    if node is null => _
    result += if result: "(" + node->value
    helper(node->left, result)
    result += "()"
    helper(node->right, result)
```

1(2(4)

```
if node->right != nullptr || node->left
    result += "("
    helper(node->left, result)
    result += ")"
if node->right
    result += "("
    helper(node->right, result)
    result += ")"
```

Idea: if left node is not null, we have to print inside, but also if right node is not null, we still have to print left's parens, even if empty