There is an error in your code:

In this line of code **if** node.right : tmp.append(node.left), you should be adding the right child node to the tmp list, but you wrote node.left instead, so only the left child node will be added to the tmp list. The correct code should be node.right .

The correct code is as follows:

```
def maxDepth(self, root: TreeNode) -> int:
    if not root: return 0
    queue, res = [root], 0
    while queue:
        tmp = []
        for node in queue:
            if node.left: tmp.append(node.left)
            if node.right: tmp.append(node.right)
        queue = tmp
        res += 1
    return res
```

The idea of this solution is: in each iteration, traverse all the nodes in the current level, and add all the nodes in the next level to the queue for the next iteration. So after each iteration, the depth res is increased by 1. Continue this process until the queue for traversal is empty, and the depth at that time is the maximum depth of the binary tree.

A.3 Analysis of Code Interpreter

Here we provide a case of comparison between Code LLaMA and Qwen-Chat. This case demonstrates the advantages of Qwen-Chat in processing tabular data and performing complex tasks.