目录

第 1 章
Tree

1.1 Maximum Depth of Binary Tree

Description

Given a binary tree, find its maximum depth.

The maximum depth is the number of nodes along the longest path from the root node down to the farthest leaf node.

Solution

```
private int maxDepth(TreeNode node) {
    if (node == null) {
        return 0;
    }
    return Math.max(maxDepth(node.left), maxDepth(node.right)) + 1;
}
```

1.2 Invert Binary Tree

Description

Invert a binary tree.

Solution I

```
public TreeNode invertTree(TreeNode root) {
    if (root == null) {
        return null;
    }

    TreeNode right = root.right;
    root.right = invertTree(root.left);
    root.left = invertTree(right);
    return root;
}
```

Solution II

```
public TreeNode invertTree(TreeNode root) {
    if (root == null) {
        return null;
    }

    Queue<TreeNode> queue = new LinkedList<TreeNode>();
    queue.offer(root);

    while (!queue.isEmpty()) {
        TreeNode node = queue.poll();

        TreeNode left = node.left;
        node.left = node.right;
        node.right = left;

        if (node.left != null) {
            queue.offer(node.left);
        }

        if (node.right != null) {
            queue.offer(node.right);
        }
    }

    return root;
}
```