226 翻转二叉树

> Label：二叉树、递归
>
> 翻转一棵二叉树。

输入：

```
     4
   /   \
  2     7
 / \   / \
1   3 6   9
```

输出：

```
     4
   /   \
  7     2
 / \   / \
9   6 3   1
```

- 深度优先 （递归）

```java
class Solution {
    public TreeNode invertTree(TreeNode root) {
        invert(root);
        return root;
    }

    private void invert (TreeNode root) {
        if (root == null)
            return;
        if (root.left == null && root.right == null)
            return;
        // 调换
        TreeNode temp = root.left;
        root.left = root.right;
        root.right = temp;

        invert(root.left);
        invert(root.right);
    }
}
```

- 广度优先

```java
class Solution {
    public TreeNode invertTree(TreeNode root) {
        if (root == null) return root;

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

        while (!queue.isEmpty()) {
            TreeNode curr = queue.poll();
            if (curr == null) {
                continue;
            }
            // 将自己的孩子节点加入队列中
            queue.add(curr.left);
            queue.add(curr.right);

            // 交换自己的左子树和右子树
            TreeNode temp = curr.left;
            curr.left = curr.right;
            curr.right = temp;
        }
        return root;
    }
}
```