144. Binary Tree Preorder Traversal

Morris traversal

无论是先序还是中序遍历，我们只要做到能判断左子树访问过了，然后访问右子树即可。所以我们先找左子树的最右节点，如果这个节点没有右节点，那么将这个节点的右节点设置为当前的节点。然后访问左子树。如果这个节点最终访问到了当前节点，说明当前节点访问过了，则访问右节点，同时把这个环去掉。如果当前节点没有左子树，则直接访问右子树时间复杂度2n，空间复杂度1

<https://leetcode.com/problems/binary-tree-preorder-traversal/solution/>

<https://zhuanlan.zhihu.com/p/101321696>

<https://blog.csdn.net/wdq347/article/details/8853371>

145. Binary Tree Postorder Traversal

方法1：递归

方法2：迭代。我的实现是用用一个额外指针指向当前节点。如果当前节点也是栈的最后一个，则表示当前节点第一次访问，如果不是，则表示这是上一个要回退的节点，利用它和栈最后一个节点的关系可以判断最后一个节点的访问状态，例如，如果这个节点是右节点的话那么就说明最后一个节点都访问过了，把最后一个节点拉出来即可。时空间复杂度n。同样可实现中序遍历

方法3：先修改前序遍历，使得其为根-右-左，然后把这个顺序反序就是后序遍历

方法4：Morris traversal。比较复杂，暂不讨论

<https://leetcode.com/problems/binary-tree-postorder-traversal/discuss/45559/My-Accepted-code-with-explaination.-Does-anyone-have-a-better-idea>

<https://blog.csdn.net/wdq347/article/details/8853371>