

二叉树遍历的非递归实现和复杂度分析

前置知识：二叉树、先序、中序、后序、栈

建议：不要跳过

- 1) 用栈实现二叉树先序遍历
- 2) 用栈实现二叉树中序遍历
- 3) 用两个栈实现二叉树后序遍历，好写但是不推荐，因为需要收集所有节点，最后逆序弹出，额外空间复杂度为 $O(n)$
- 4) 用一个栈实现二叉树后序遍历
- 5) 遍历二叉树复杂度分析：
 - a. 时间复杂度 $O(n)$ ，递归和非递归都是每个节点遇到有限几次，当然 $O(n)$
 - b. 额外空间复杂度 $O(h)$ ，递归和非递归都需要二叉树高度 h 的空间来保存路径，方便回到上级去
 - c. 存在时间复杂度 $O(n)$ ，额外空间复杂度 $O(1)$ 的遍历方式：*Morris*遍历
 - d. *Morris*遍历比较难，也比较冷门，会在【扩展】课程里讲述

关于递归更多的内容会在【必备】课程里继续

二叉树更多更难的题会在【必备】课程里继续