

Morris遍历

前置知识

讲解017、讲解018 - 递归方式遍历二叉树、非递归方式遍历二叉树

之前讲过的二叉树三种序的遍历

不管是递归还是非递归的方式，时间复杂度为 $O(n)$ ，额外空间复杂度为 $O(\text{树高})$

Morris遍历解决的问题

二叉树三种序的遍历，时间复杂度为 $O(n)$ ，额外空间复杂度为 $O(1)$

核心在于利用二叉树自身的空闲指针

大厂笔试、比赛时，一般不会用到

大厂面试时，一道普通的二叉树题目，可以先给出普通解法，

然后和面试官聊Morris遍历的解法，可以增加印象分(装逼用)

工程上很多场合，内存空间非常有限，遍历二叉树时可以使用这种遍历

Morris遍历

Morris遍历的理解核心

没有左子树的节点只到达一次，有左子树的节点会到达两次
利用左子树最右节点的右指针状态，来标记是第几次到达

Morris遍历的过程，课上重点图解

- 1，开始时 cur 来到头节点， cur 为空时过程停止
- 2，如果 cur 没有左孩子， cur 向右移动
- 3，如果 cur 有左孩子，找到 cur 左子树的最右节点 $mostRight$
 - A，如果 $mostRight$ 的右指针指向空，让其指向 cur ，然后 cur 向左移动
 - B，如果 $mostRight$ 的右指针指向 cur ，让其指向 $null$ ，然后 cur 向右移动

额外空间复杂度很明显是 $O(1)$ ，但是时间复杂度依然为 $O(n)$ ，课上重点图解

Morris遍历

题目1

Morris遍历实现先序和中序遍历

测试链接：<https://leetcode.cn/problems/binary-tree-preorder-traversal/>

测试链接：<https://leetcode.cn/problems/binary-tree-inorder-traversal/>

Morris遍历

题目2

Morris遍历实现后序遍历

测试链接：<https://leetcode.cn/problems/binary-tree-postorder-traversal/>

Morris遍历实现后序遍历，需要树上链表的翻转，课上重点图解

Morris遍历

重要提醒

用递归方式遍历二叉树能解决的问题，并不一定都可以用*Morris*遍历解决

递归方式遍历二叉树，每个节点都能到达**3**次，
所以每棵子树都能收集左子树信息、右子树信息，然后整合出整棵子树的信息

*Morris*方式遍历二叉树，每个节点最多到达**2**次，
所以信息整合的强度不及递归的方式

具体哪些问题能被*Morris*遍历解决，需要具体问题具体分析

如果必须回到一个节点**3**次，问题才能得到解决，那么这样的问题无法用*Morris*遍历解决

Morris遍历

题目3

Morris遍历判断搜索二叉树

测试链接：<https://leetcode.cn/problems/validate-binary-search-tree/>

Morris遍历

题目4

Morris遍历求二叉树最小高度

测试链接：<https://leetcode.cn/problems/minimum-depth-of-binary-tree/>

Morris遍历

题目5

Morris遍历求两个节点的最低公共祖先

测试链接：<https://leetcode.cn/problems/lowest-common-ancestor-of-a-binary-tree/>