

### 8.1.4 二叉树遍历算法

链表是一种线性结构，它允许我们用指针 `next` 去顺序遍历节点。因为树是一种非线性结构，不能用类似的遍历算法。我们只能从各种遍历算法中进行选择，常用的有前序遍历、中序遍历和后序遍历。这些算法的出发点都是二叉树的递归结构。

作为一种递归结构，二叉树的每个节点由它的值和它的左、右子树定义。遍历树的过程中，对每个节点包含 3 步操作：访问节点自身并执行一些任务(N)，递归下降到左子树(L)，递归下降到右子树(R)。下降到子树以后，遍历行进到左子节点或右子节点，它们是相应子树的根节点，算法此时对当前节点重复同样的三步动作。一直到遇到空树(指针=NULL)时，递归下降过程终止。各种递归算法的不同之处在于它们对每个节点执行这 3 步操作的顺序，共有 NLR、LNR、LRN、NRL、RNL 和 RLN 六种顺序，其中前 3 种是按先左后右的次序遍历根的两棵子树，后 3 种是按先右后左的顺序遍历子树。由于两者对称，一般我们只讨论前 3 种遍历方案。

#### 1. 中序遍历(Inorder Scan)

在中序遍历中，对节点实施的第一个动作是转到左子树以便遍历该子树中的节点。在递归遍历完左子树后，中序遍历对节点实施第二个动作，处理节点数据值，最后一个动作是递归遍历右子树。中序遍历的操作次序如下：

- (1) 遍历左子树(向左)。
- (2) 访问节点。
- (3) 遍历右子树(向右)。

这种遍历方法被称为 LNR(左子树、节点、右子树)。

#### 2. 后序遍历(Postorder Scan)

后序遍历将对节点的访问推迟到递归访问完左子树和右子树以后。这种操作次序被称为 LRN 遍历(左子树、右子树、节点)。

- (1) 遍历左子树(向左)。
- (2) 遍历右子树(向右)。
- (3) 访问节点。

#### 3. 前序遍历(Preorder Scan)

它规定先访问节点，然后再扫描左右分支(NLR)。

显然，前缀限定词“前”、“中”、“后”分别表示对节点的访问发生在何时。

#### 迭代层次遍历(Iterative Level-Order Scan)

迭代层次遍历，又称广度优先遍历，按层次来访问元素，首先是根节点，然后是第一层的子节点，依次类推。层次遍历是迭代的过程，它用一个队列作为中间存储容器。初始时刻，根节点进入队列。重复的迭代动作包含从队列头弹出节点，对该节点执行一些动作，之后将其子节点压入队列尾部。因为兄弟节点是在访问其父节点时加入队列的，所以在同一层的兄弟节点会以连续迭代顺序退出队列。