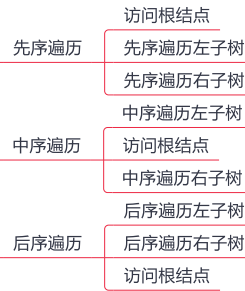


5.3 二叉树的遍历和线索二叉树

二叉树的遍历



时间复杂度 $O(n)$

递归算法和非递归算法的转换：利用栈进行实现

层次遍历

利用队列实现

实现

- 先将二叉树根结点入队，然后出队，访问出队结点
- 若它有左子树，则将左子树根结点入队
- 若它有右子树，则将右子树根结点入队
- 然后出队，访问出队结点.....如此反复，直至队列为空

由遍历序列构造二叉树

二叉树的先序（后序）序列和中序序列可以唯一地确定一棵二叉树

层序遍历与中序（后序）可以确定一颗二叉树

线索二叉树

基本概念

线索二叉树的基本概念

遍历二叉树是以一定的规则将二叉树中的结点排列成一个线性序列，从而得到几种遍历序列

使得该序列中的每个结点（第一个和最后一个结点除外）都有一个直接前驱和直接后继

规定

若无左子树，令lchild指向其前驱结点

若无右子树，令rchild指向其后继结点

结点结构

| lchild | ltag | data | rtag | rchild |
|--------|------|------|------|--------|
|--------|------|------|------|--------|

标志域的含义

ltag 0 lchild域指示结点的左孩子

1 lchild域指示结点的前驱

rtag 0 rchild域指示结点的右孩子

1 rchild域指示结点的后继

概念

二叉树的线索化是将二叉链表中的空指针改为指向前驱或后继的线索

而前驱或后继的信息只有在遍历时才能得到，因此线索化的实质就是遍历一次二叉树

中序线索二叉树的构造

中序线索二叉树的建立

附设指针 pre指向刚刚访问过的结点，指针p指向正在访问的结点，即pre指向p的前驱

在中序遍历的过程中

检查p的左指针是否为空，若为空就将它指向pre

检查pre的右指针是否为空，若为空就将它指向p

中序线索二叉树的遍历

若其右标志为"1"，则右链为线索，指示其后继，否则遍历右子树中第一个访问的结点（右子树中最左下的结点）为其后继

先序线索二叉树

先序序列为ABCDF，然后依次判断每个结点的左右链域，如果为空则将其改造为线索

结点A、B均有左右孩子；结点C无左孩子，将左链域指向前驱B，无右孩子，将右链域指向后继D

结点D无左孩子，将左链域指向前驱C，无右孩子，将右链域指向后继F

结点F无左孩子，将左链域指向前驱D，无右孩子，也无后继故置空

后序线索二叉树

后序序列为CDBFA，结点C无左孩子，也无前驱故置空，无右孩子，将右链域指向后继D

结点D无左孩子，将左链域指向前驱C，无右孩子，将右链域指向后继B

结点F无左孩子，将左链域指向前驱B，无右孩子，将右链域指向后继A