

4.1.3 二叉树的遍历

问题的提出

顺着某一条搜索路径巡访二叉树中的结点，使得每个结点均被访问一次，而且仅被访问一次。

常见的遍历方式有：

递归遍历

层次遍历

非递归遍历

先序遍历算法：

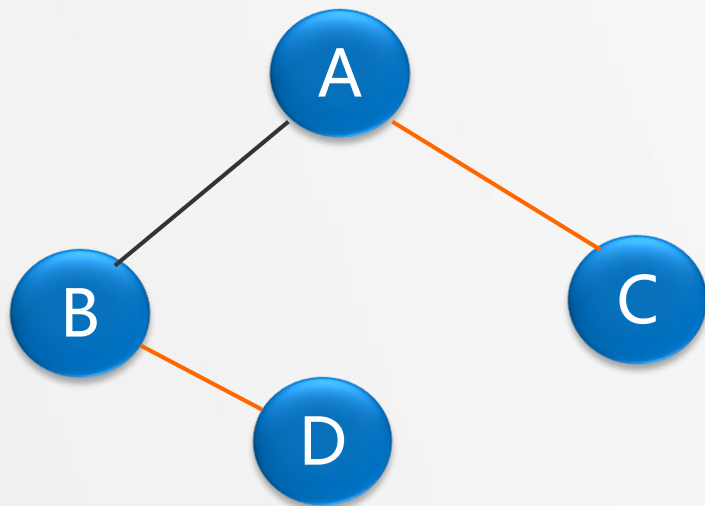
若二叉树非空，则，

- (1) 访问根结点；
- (2) 先序遍历左子树；
- (3) 先序遍历右子树。

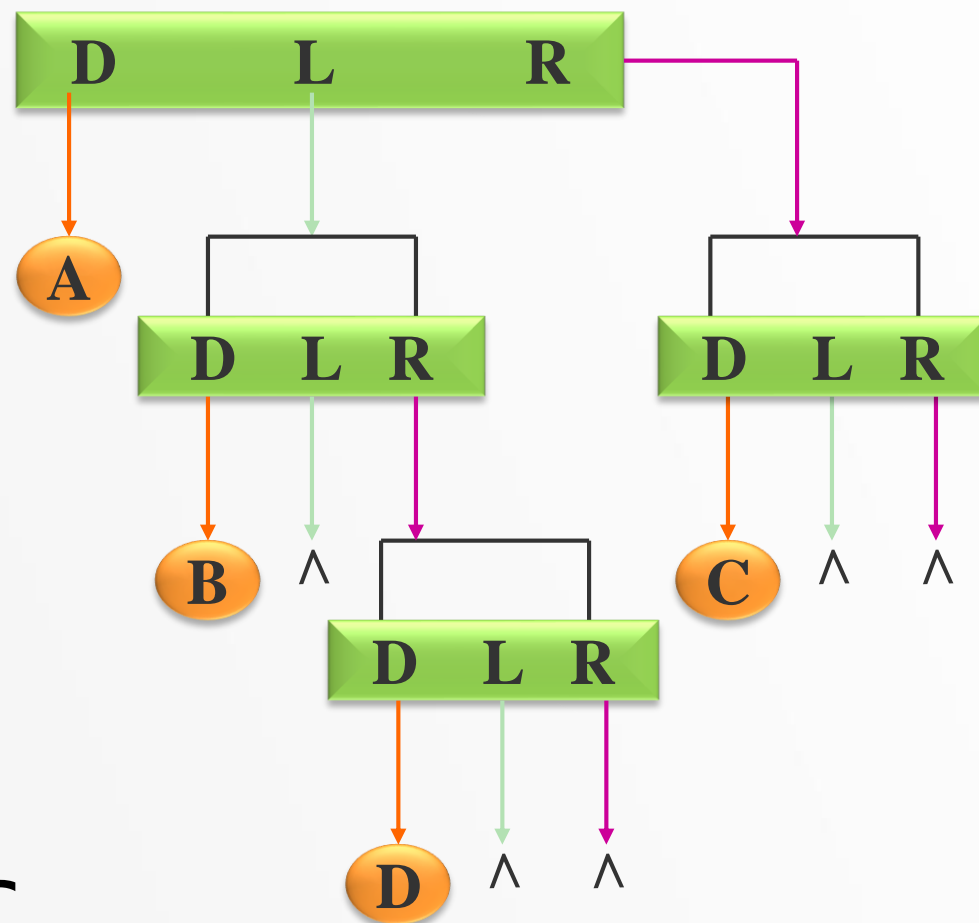
二叉树的递归遍历 | 先序遍历DLR



二叉树先序遍历



先序遍历序列: A B D C



中序遍历算法：

若二叉树非空，则，

- (1) 中序遍历左子树；
- (2) 访问根结点；
- (3) 中序遍历右子树。

圖

```
graph TD; A((A)) --- B((B)); A((A)) --- C((C)); B((B)) --- D((D));
```

The diagram illustrates a 3D tree structure. The root node is a green rectangular box labeled L, D, R. It has three outgoing arrows: a green arrow to the left, an orange arrow to the middle, and a purple arrow to the right. The green arrow points to a green rectangular box labeled L, D, R. The orange arrow points to an orange oval labeled A. The purple arrow points to a green rectangular box labeled L, D, R. The green rectangular box labeled L, D, R (under the green arrow) has three outgoing arrows: a green arrow to the left pointing to a black symbol \wedge , an orange arrow to the middle pointing to an orange oval labeled B, and a purple arrow to the right pointing to a green rectangular box labeled L, D, R. The green rectangular box labeled L, D, R (under the purple arrow) has three outgoing arrows: a green arrow to the left pointing to a black symbol \wedge , an orange arrow to the middle pointing to an orange oval labeled C, and a purple arrow to the right pointing to a black symbol \wedge . The green rectangular box labeled L, D, R (under the orange arrow from the middle box) has three outgoing arrows: a green arrow to the left pointing to a black symbol \wedge , an orange arrow to the middle pointing to an orange oval labeled D, and a purple arrow to the right pointing to a black symbol \wedge .

后序遍历算法：

若二叉树非空，则，

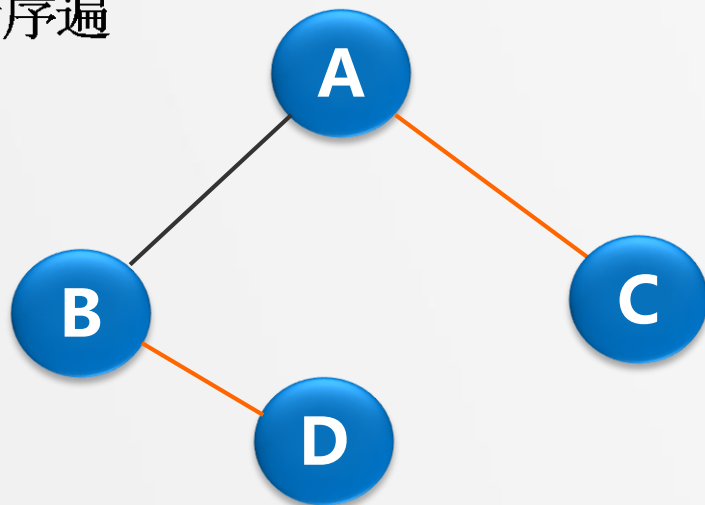
- (1) 后序遍历左子树；
- (2) 后序遍历右子树；
- (3) 访问根结点。



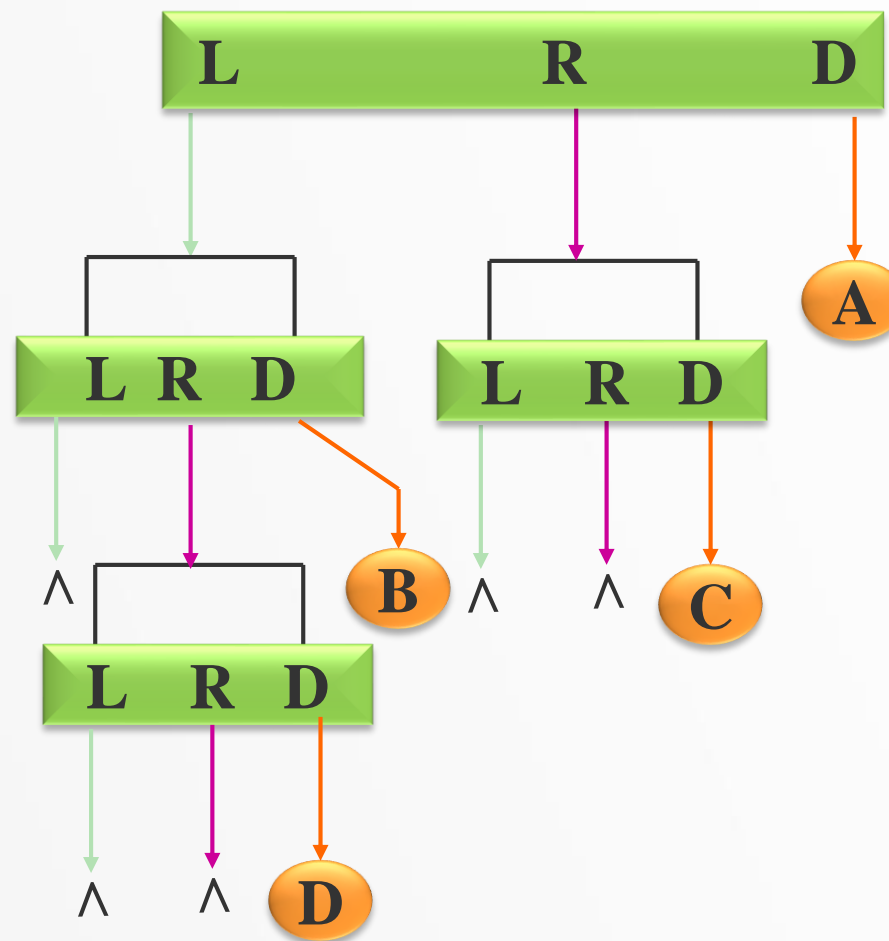
二叉树的递归遍历 | 后序遍历DLR



二叉树后序遍历



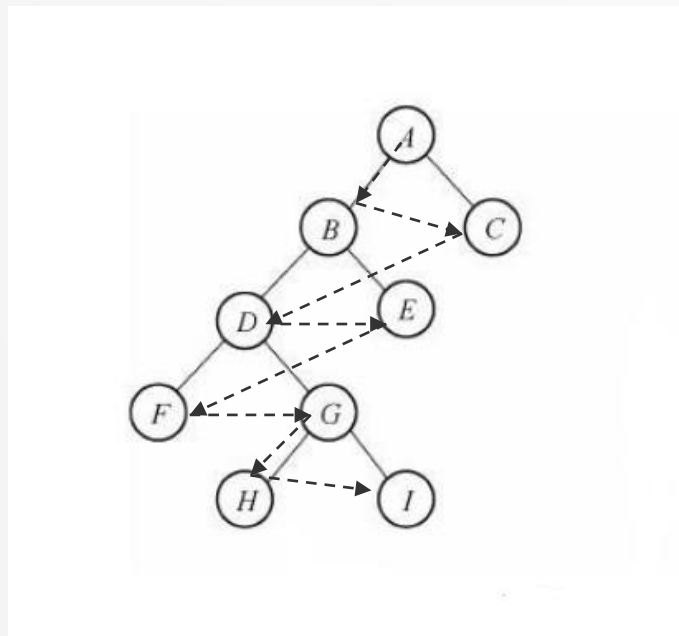
后序遍历序列: D B C A



讨论

二叉树是否可以来进行从右到左的遍历?如果可以, 以上3个例子的遍历序列是什么?

二叉树的层次遍历



层次遍历

4.1.4 二叉树遍历算法的应用举例

- 1、建立二叉树(先序和中序)
- 2、统计二叉树中叶子结点的个数(先序遍历)
- 3、求二叉树的深度(后序遍历)

思考：

- 1、仅知道二叉树的先序序列能否唯一确定一颗二叉树？
- 2、知道同一颗树的两种不同序列能否唯一的确定一颗二叉树？
例如先序和中序、先序和后序、后序和中序、层次和中序.....

1. 由先序和中序遍历序列建立二叉树

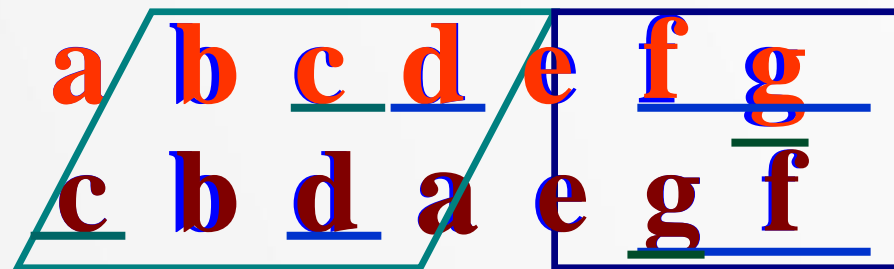
二叉树的先序序列



二叉树的中序序列



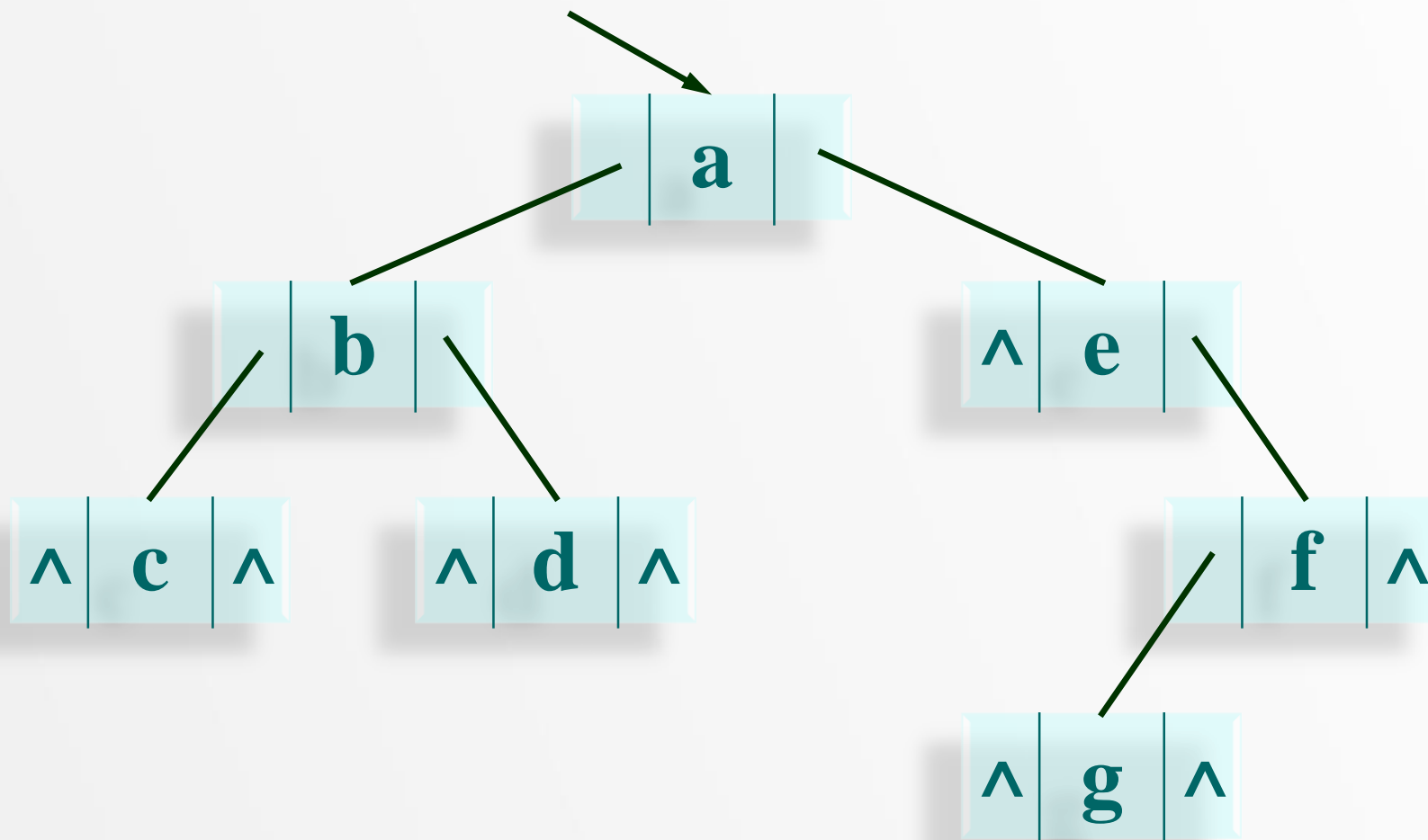
例如:



先序序列
中序序列



先序+中序=二
叉树



2、二叉树中叶结点统计

先序(中序或后序)遍历二叉树，在遍历过程中查找叶结点，将算法中“访问结点”的操作改为：**判叶结点**。



统计叶子结点
数目

3. 二叉树的深度

空树：深度=0

左右子树为空：深度=1

其它：深度=1+max(左子树深度，右子树深度)

遍历



二叉树深度

作业

设一棵二叉树其中序和后序遍历为

中序: BDCEAFHG 后序: DECBHGFA

画出这棵二叉树的逻辑结构,并给出先序遍历结果