

数据结构（C语言版）（第2版）



树和二叉树

线索二叉树

主讲教师：汪红松



教学内容 Contents

- 
- 1 树和二叉树的定义
 - 2 二叉树的性质和存储结构
 - 3 遍历二叉树
 - 4 线索二叉树
 - 5 树和森林
 - 6 哈夫曼树及其应用

▶▶▶一、线索化二叉树

普通二叉树只能找到结点的左右孩子信息，而该结点的直接前驱和直接后继只能在遍历过程中获得。

若将遍历后对应的有关前驱和后继预存起来，则从第一个结点开始就能很快“顺藤摸瓜”而遍历整个树。

例如中序遍历结果：B D C E A F H G，实际上已将二叉树转为线性排列，显然具有唯一前驱和唯一后继！

▶▶▶ 一、线索化二叉树

如何保存这类信息？

增加两个域：
fwd和bwd；



利用空链域
($n+1$ 个空链域)

▶▶▶ 一、线索化二叉树

1)

若结点有左子树，则lchild指向其左孩子；
否则， lchild指向其直接前驱(即线索)；

2)

若结点有右子树，则rchild指向其右孩子；
否则， rchild指向其直接后继(即线索)。

为了避免混淆，增加两个标志域

lchild

LTag

data

RTag

rchild

▶▶▶ 一、线索化二叉树 1.术语

结点结构

```
struct ThrNode  
{  
    TElemType data;  
    struct BiThrNode *lchild, *rchild;  
    int ltag, rtag;  
};
```

线索
指向结点前驱和
后继的指针

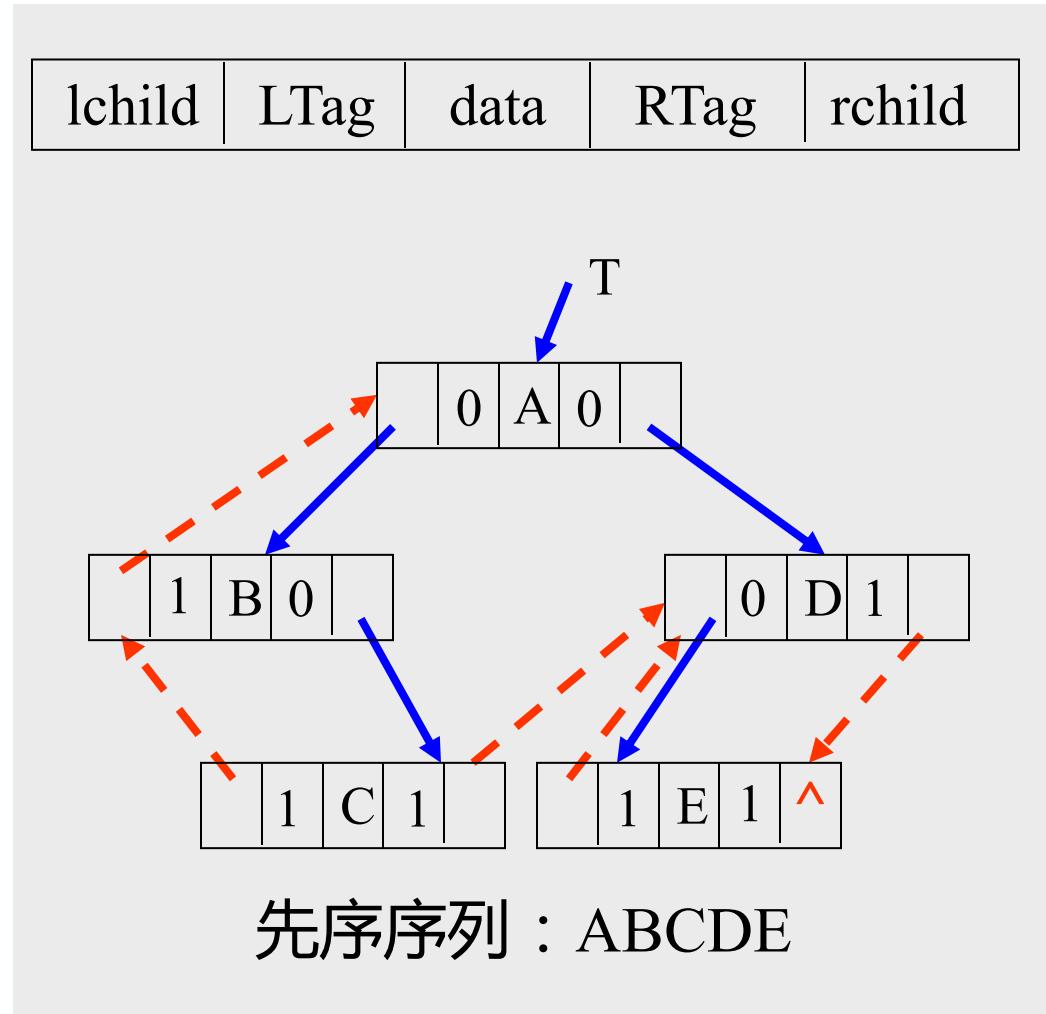
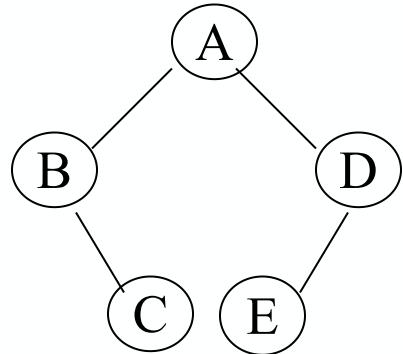


线索链表
加上线索二叉链表

线索二叉树
加上线索的二叉
树（图形式样）

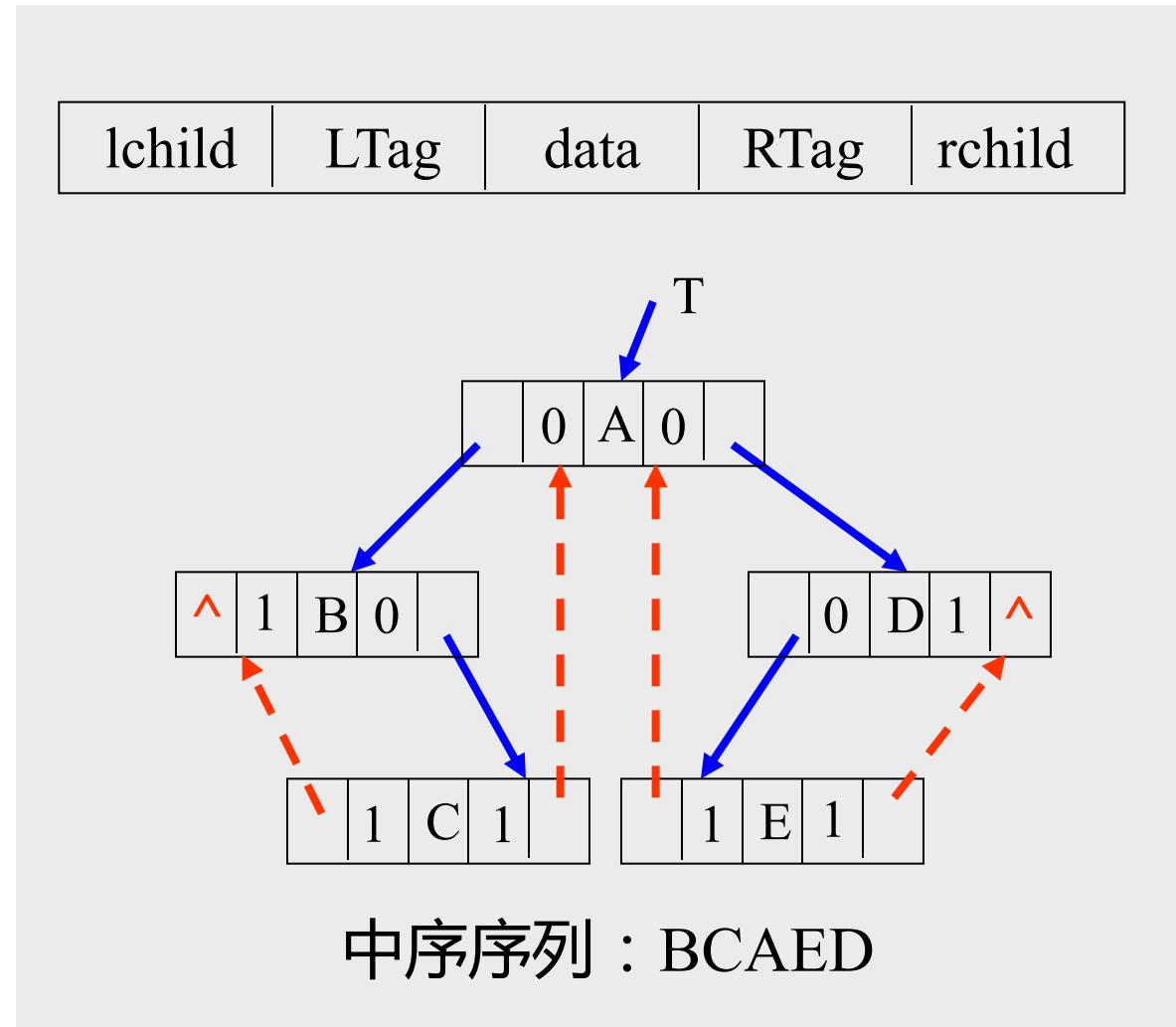
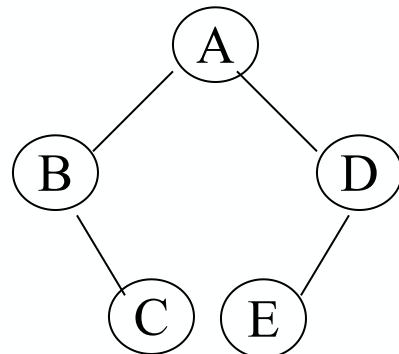
线索化
对二叉树以某种次
序遍历使其变为线
索二叉树的过程

一、线索化二叉树 2.先序线索二叉树



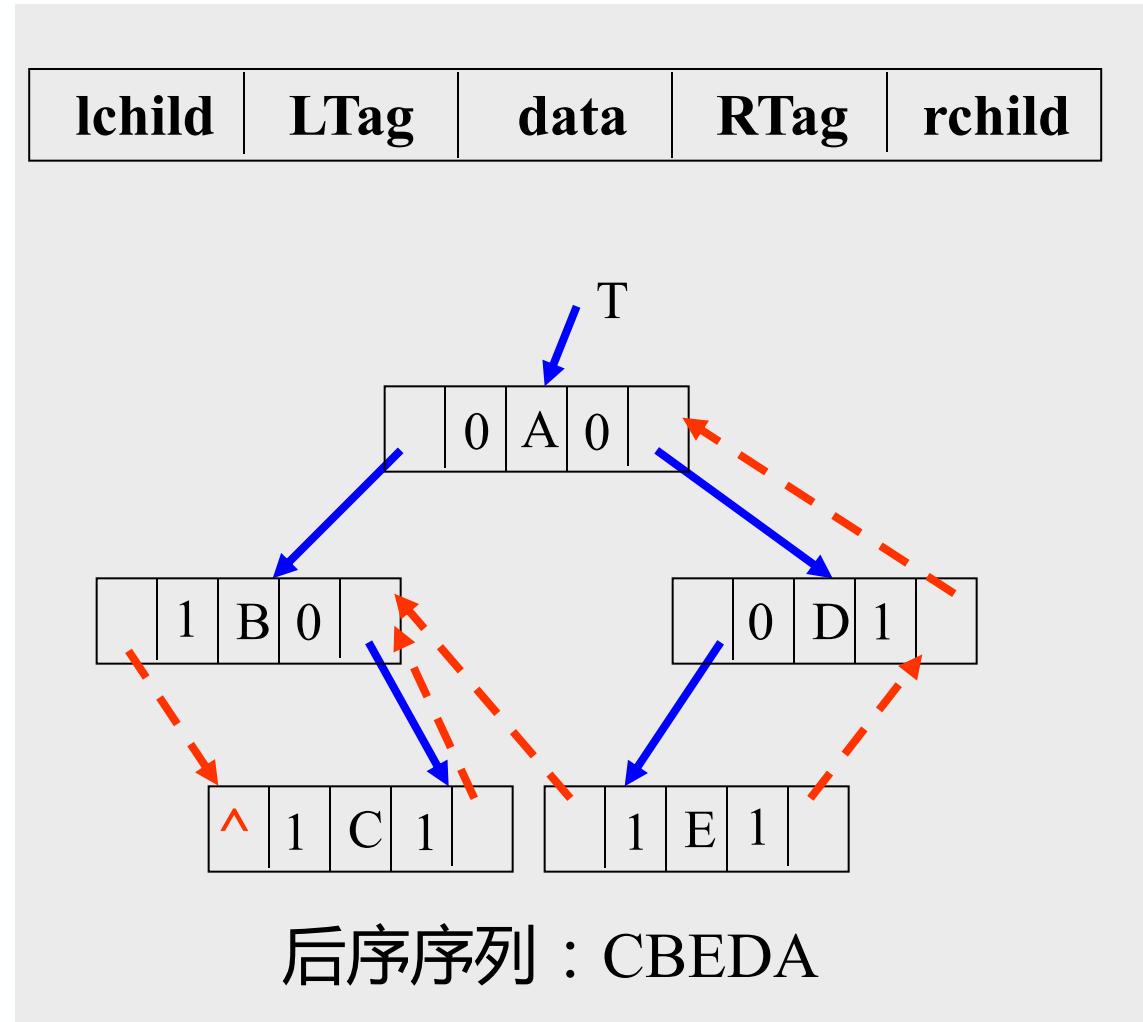
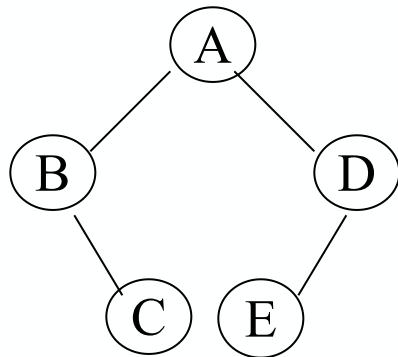
一、线索化二叉树

3. 中序线索二叉树



一、线索化二叉树

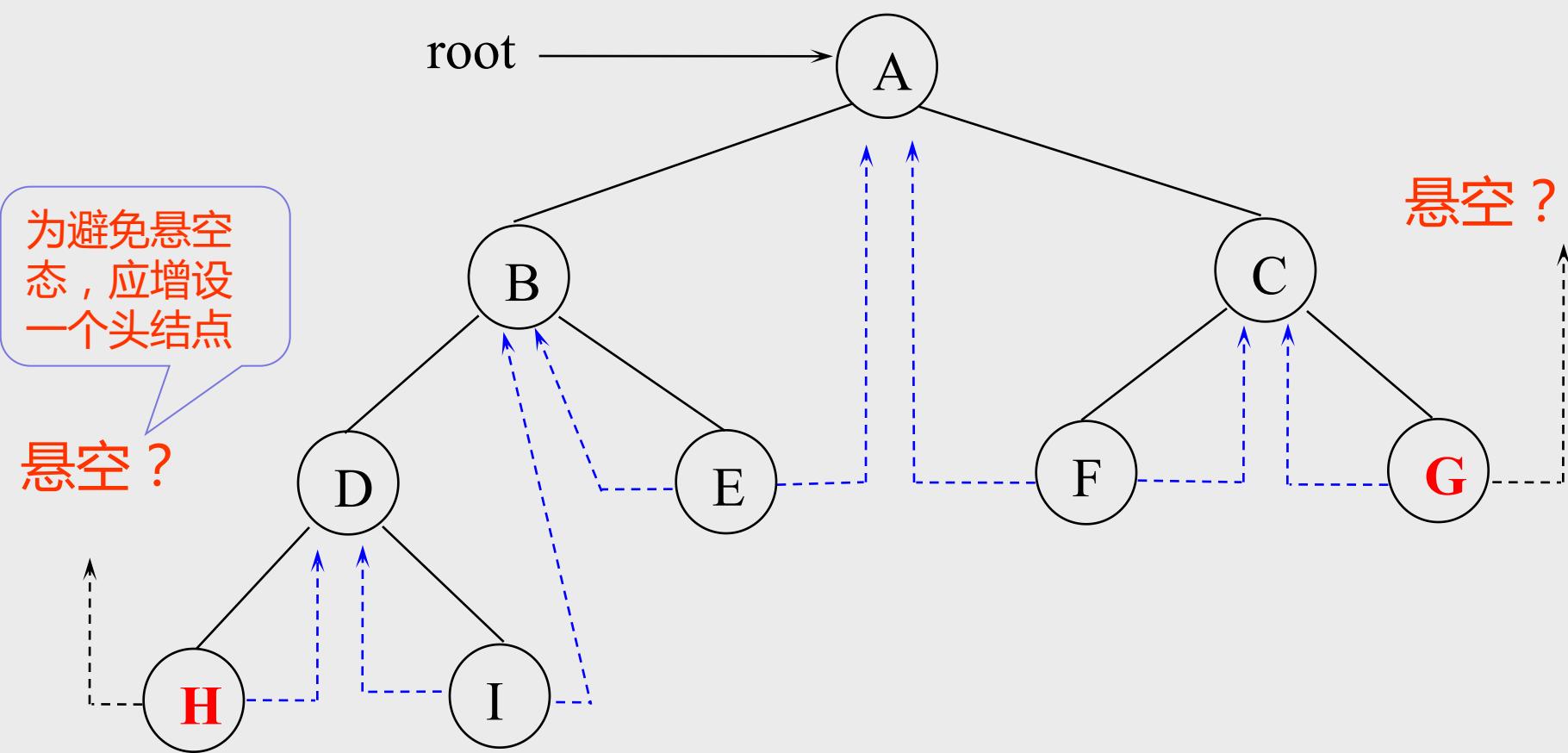
4.后序线索二叉树



▶▶▶ 练习

画出以下二叉树对应的中序线索二叉树。

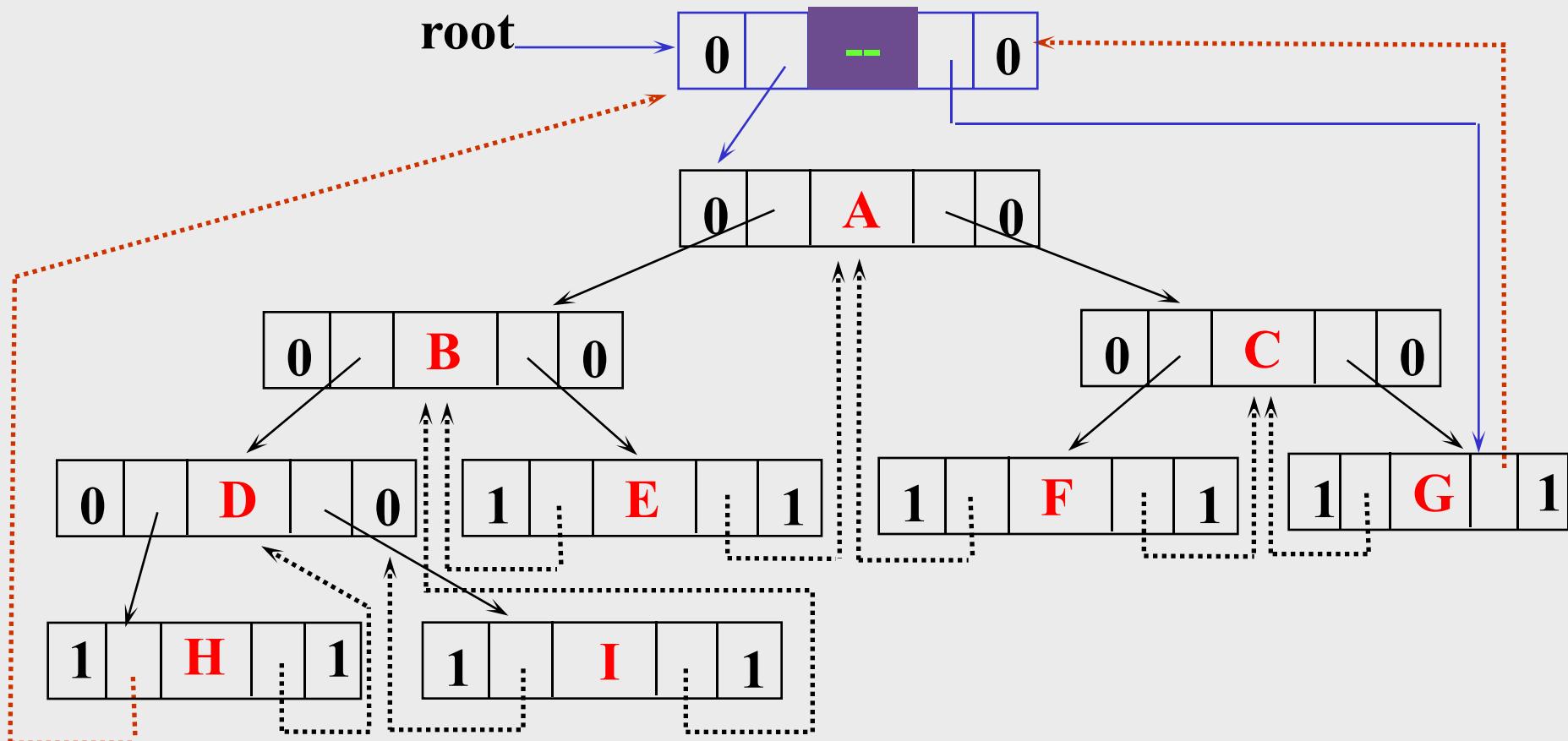
该二叉树中序遍历结果为: H, D, I, B, E, A, F, C, G



练习

为中序线索二叉链表增加头结点。

注：此图中序遍历结果为： H, D, I, B, E, A, F, C, G



二、构造中序线索二叉链表

构造线索二叉树

建立线索链表，实质上就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该二叉树时才能得到。

建立二叉链表



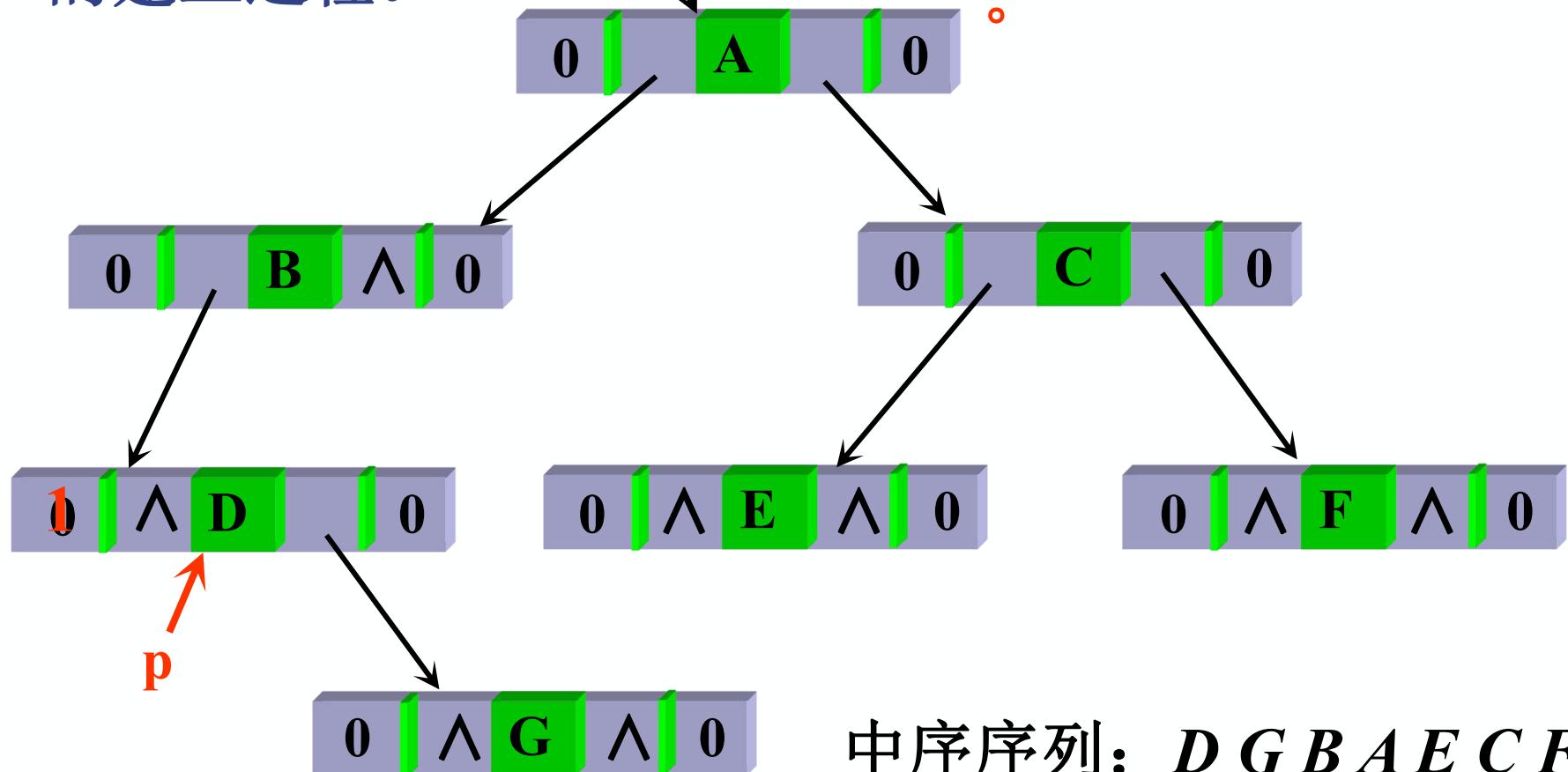
遍历二叉树，将空指针改为线索

二、构造中序线索二叉链表

中序线索链表
的建立过程。

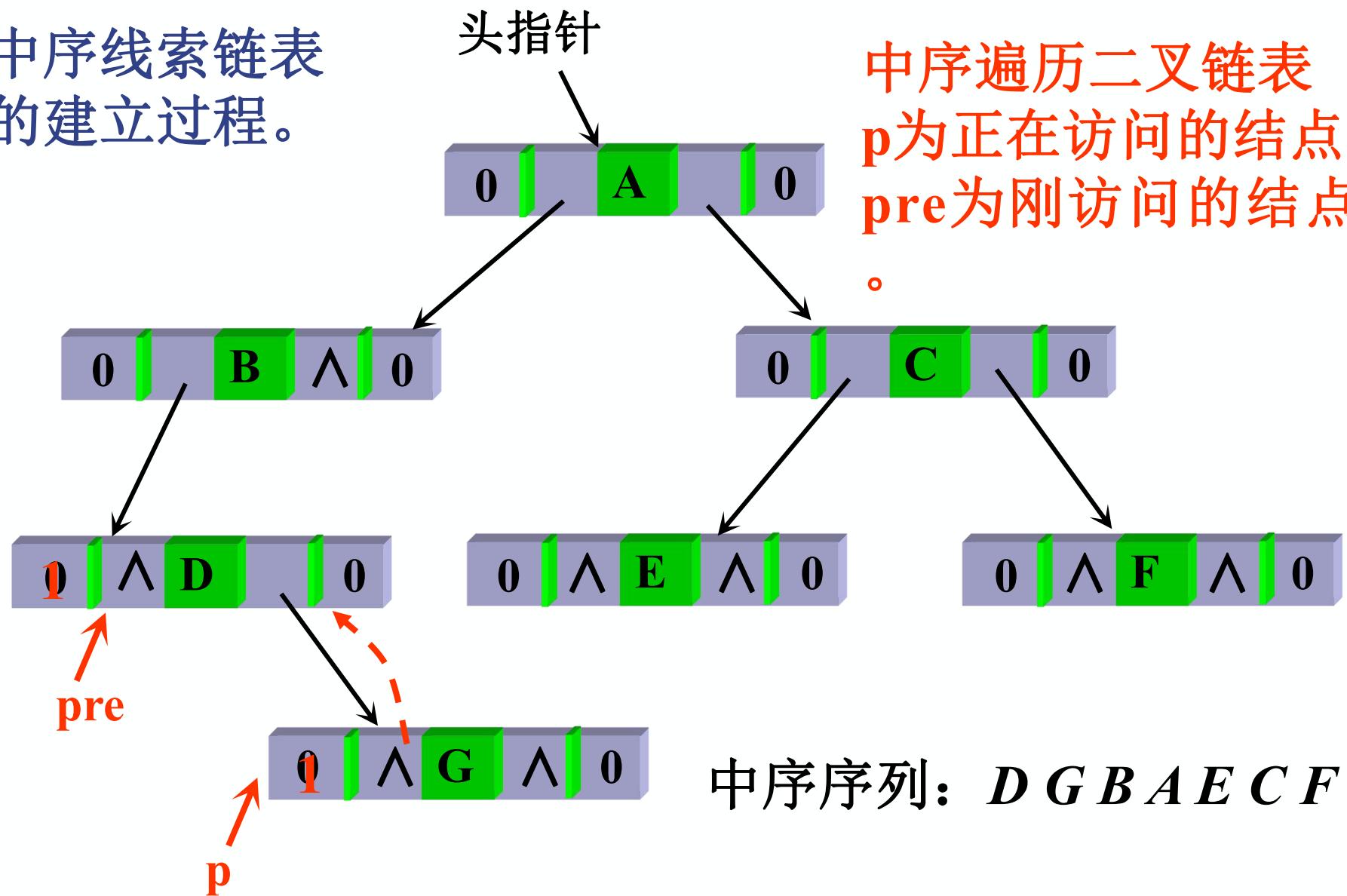
头指针

已经建立起二叉链表
。



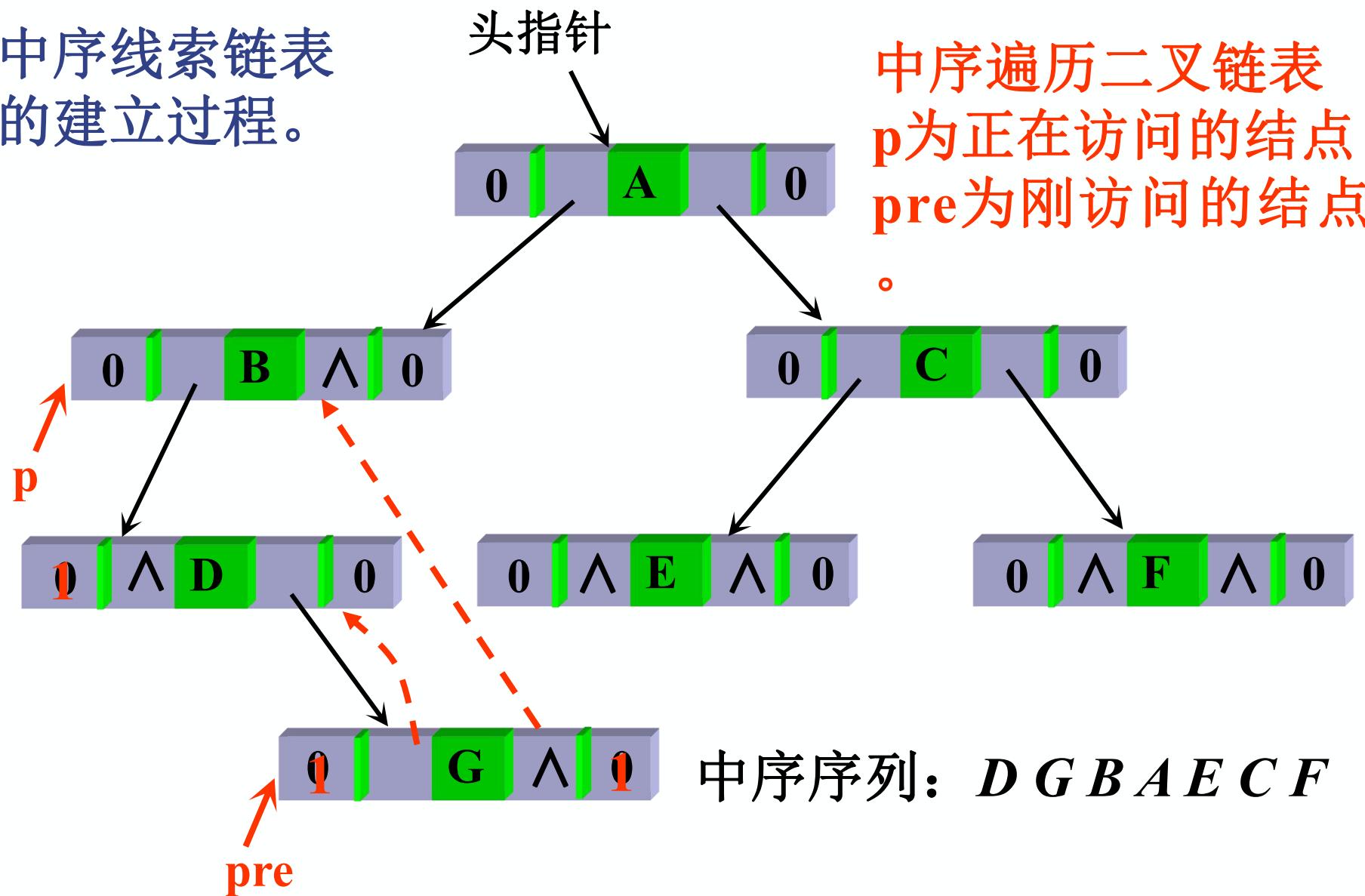
二、构造中序线索二叉链表

中序线索链表
的建立过程。



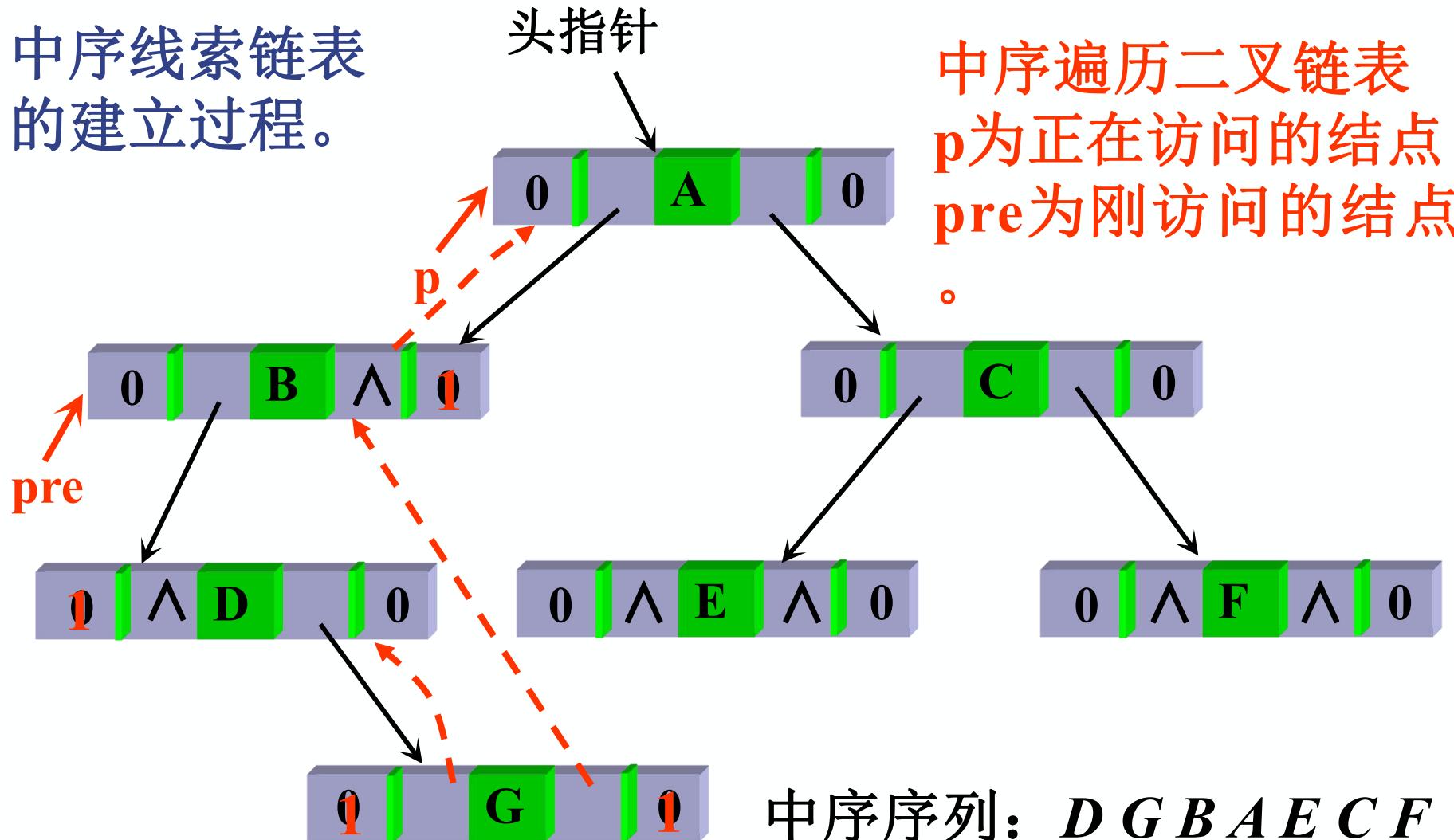
二、构造中序线索二叉链表

中序线索链表
的建立过程。



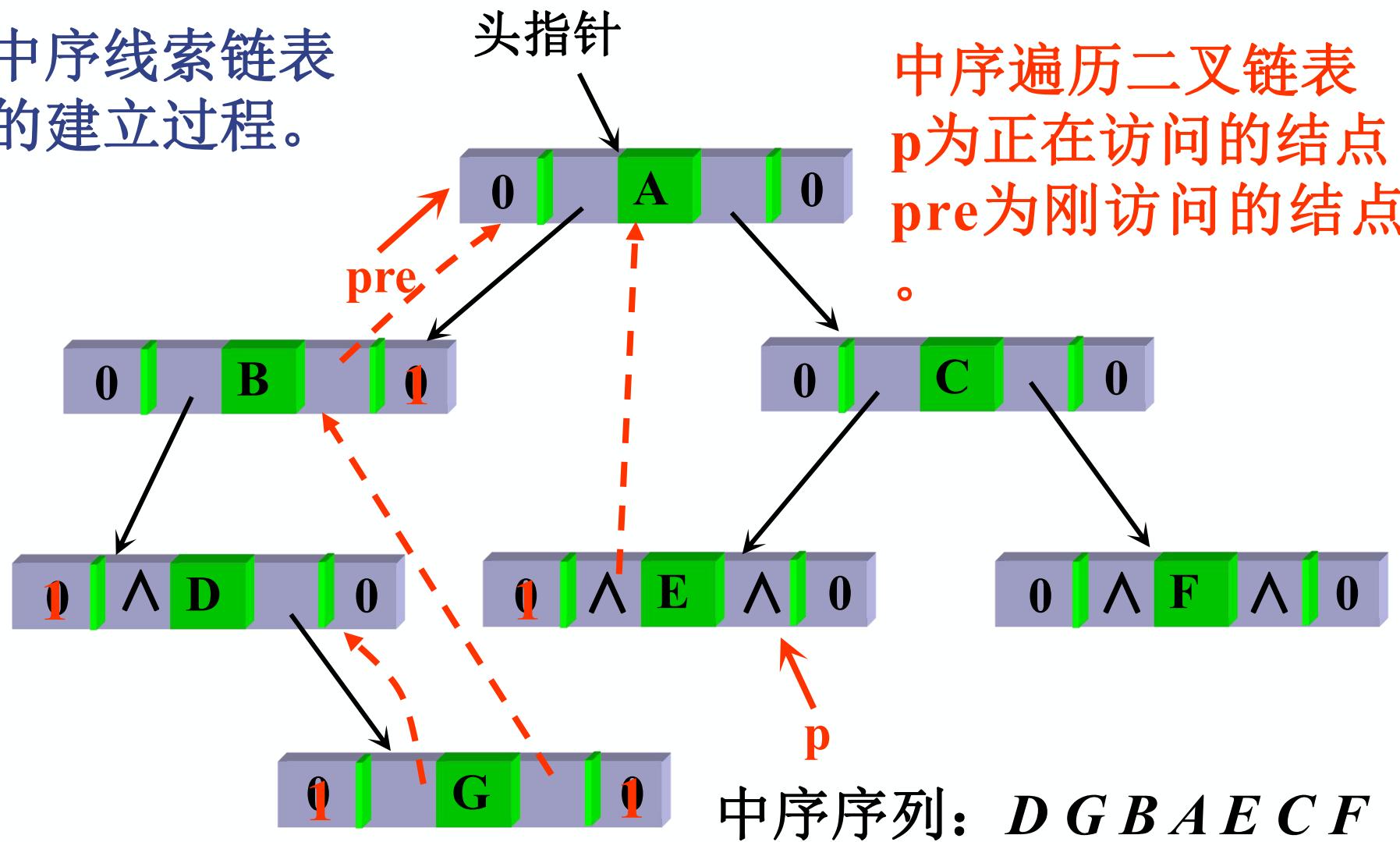
二、构造中序线索二叉链表

中序线索链表
的建立过程。



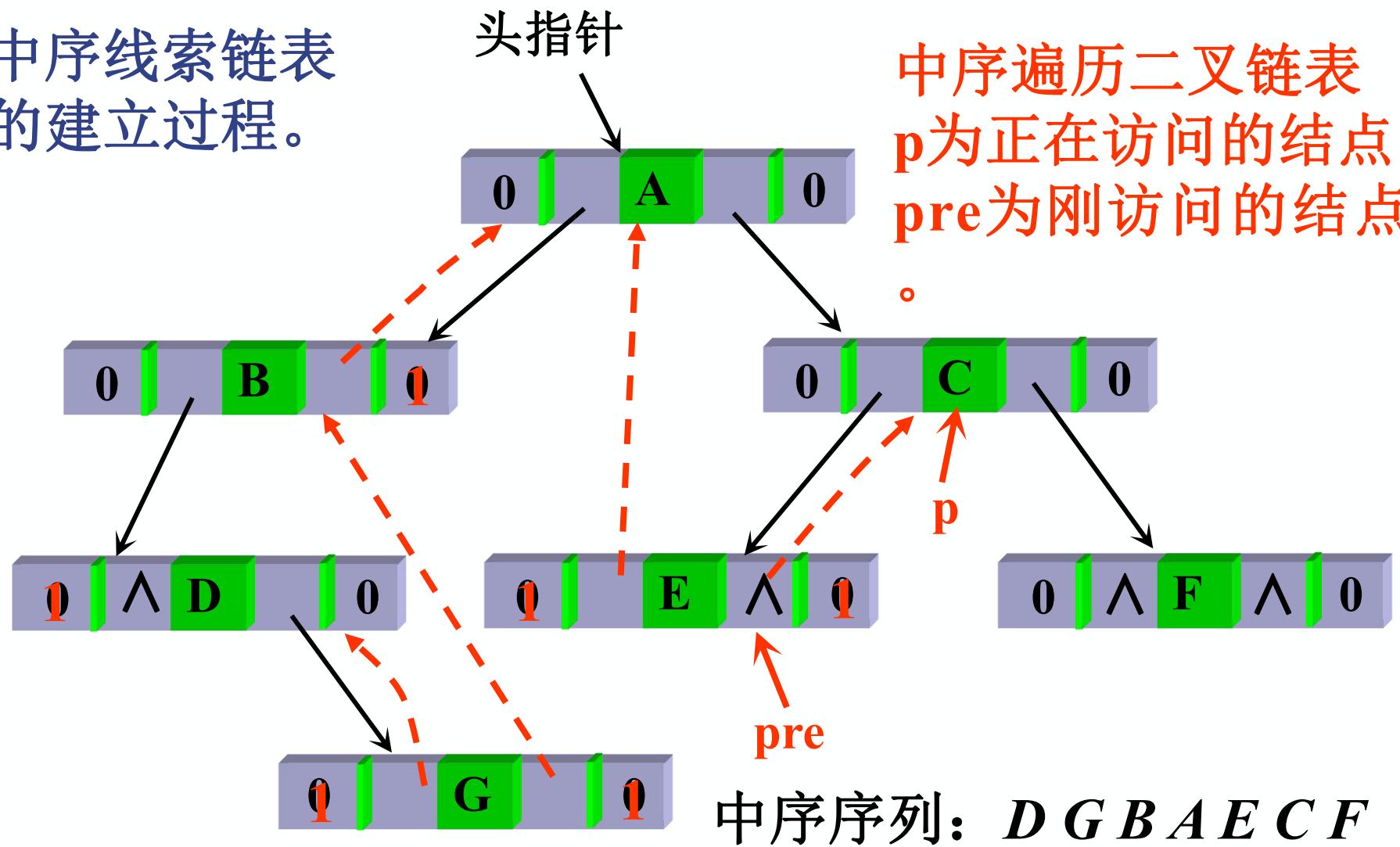
二、构造中序线索二叉链表

中序线索链表
的建立过程。



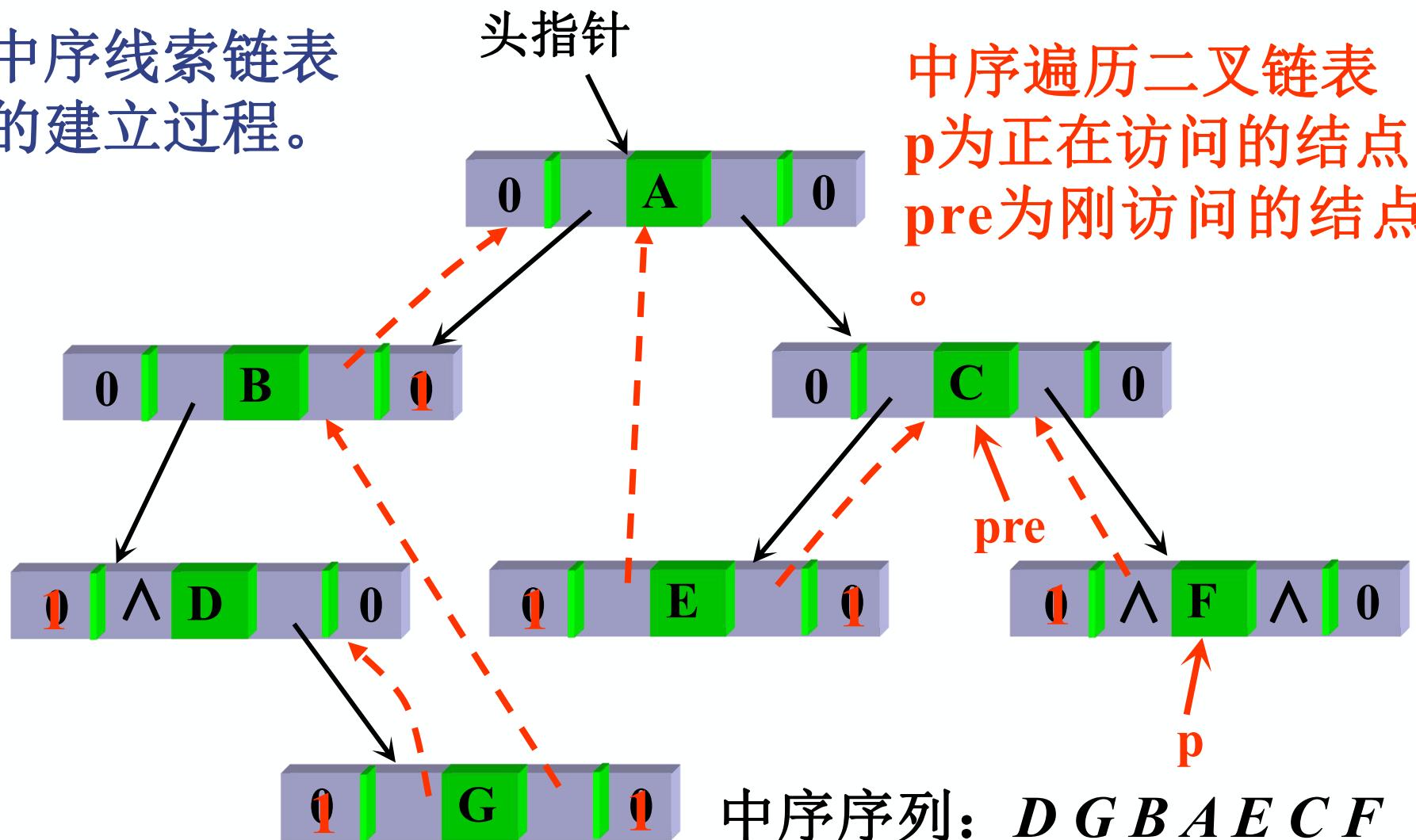
二、构造中序线索二叉链表

中序线索链表
的建立过程。



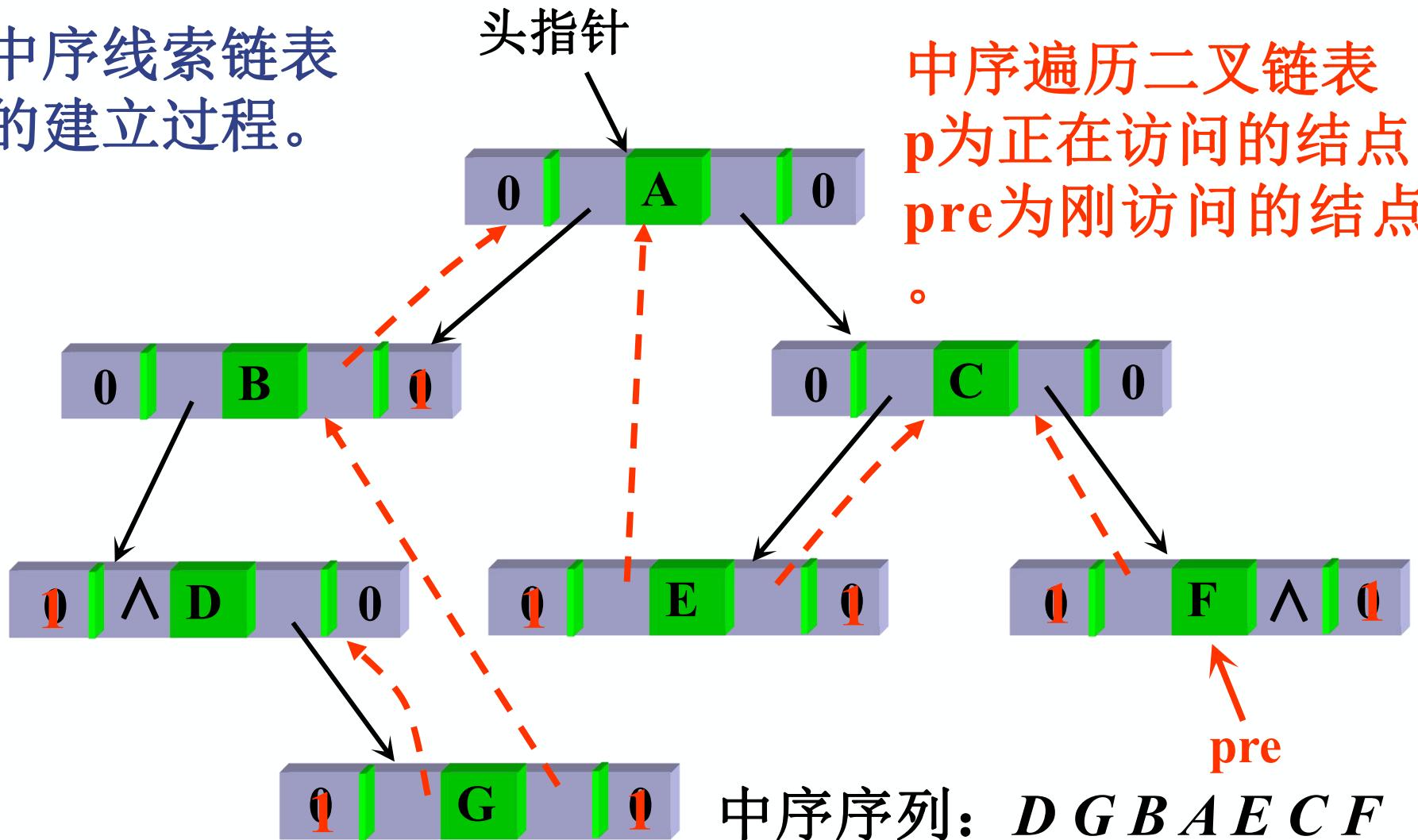
二、构造中序线索二叉链表

中序线索链表
的建立过程。



二、构造中序线索二叉链表

中序线索链表
的建立过程。



二、构造中序线索二叉链表

1. 中序线索链表的建立

在遍历过程中，访问当前结点root的操作为：

- (1) 如果root的左、右指针域为空，则将相应标志置1；
- (2) 若root的左指针域为空，则令其指向它的前驱，这需要设指针pre始终指向刚刚访问过的结点，显然pre的初值为NULL；若pre的右指针域为空，则令其指向它的后继，即当前访问的结点root；
- (3) 令pre指向刚刚访问过的结点root。

二、构造中序线索二叉链表

2. 中序线索链表的建立

- (1) 建立二叉链表，将每个结点的左右标志置为0；
- (2) 遍历二叉链表，建立线索；
 - ①如果二叉链表root为空，则空操作返回；
 - ②对root的左子树建立线索；
 - ③对根结点root建立线索；
 - I 若root没有左孩子，则为root加上前驱线索；
 - II 若root没有右孩子，则将root右标志置为1；
 - III 若结点pre右标志为1，则为pre加上后继线索；
 - IV 令pre指向刚刚访问的结点root；
 - ④对root的右子树建立线索。

三、遍历线索二叉树

1. 中序遍历线索二叉树

算法思想：

(1) 从根结点出发沿左指针向下，到达最左下结点*p，它是中序的第一个结点，访问*p。

(2) 反复查找当前结点*p的后继结点，直至遍历结束；

若p->RTag为1，则其后继结点的指针即为p->rchild；

否则，其后继为结点*p的右子树的最左下结点；

访问找到这个后继结点。

三、遍历线索二叉树

1. 中序遍历线索二叉树

```
void InOrderTraverse Thr (BiThrTree T)
{
    p=T->lchild;           //p 指向根结点
    while(p!=T){            //空树或遍历结束时,p= =T
        while(p->LTag==0) p=p->lchild; //沿左孩子向下
        cout<<p-> data;           //访问其左子树为空的结点
        while (p->RTag==1&&p->rchild!=T) {
            p=p->rchild;cout<<p-> data; //沿右线索访问后继结点
        }
        p=p->rchild;
    }
}
```



小结

1. 线索二叉树的概念
2. 构造线索二叉树的方法
3. 遍历线索二叉树的算法步骤