Data Structures

Ch5

树和二叉树

Trees & Binary Trees

2024 年 10 月 29 日

学而不厌 诲人不倦

Chapter 5　树和二叉树

5.8 扩展与提高

5-8-1　非递归中序遍历二叉树

5.8.1 非递归中序遍历二叉树

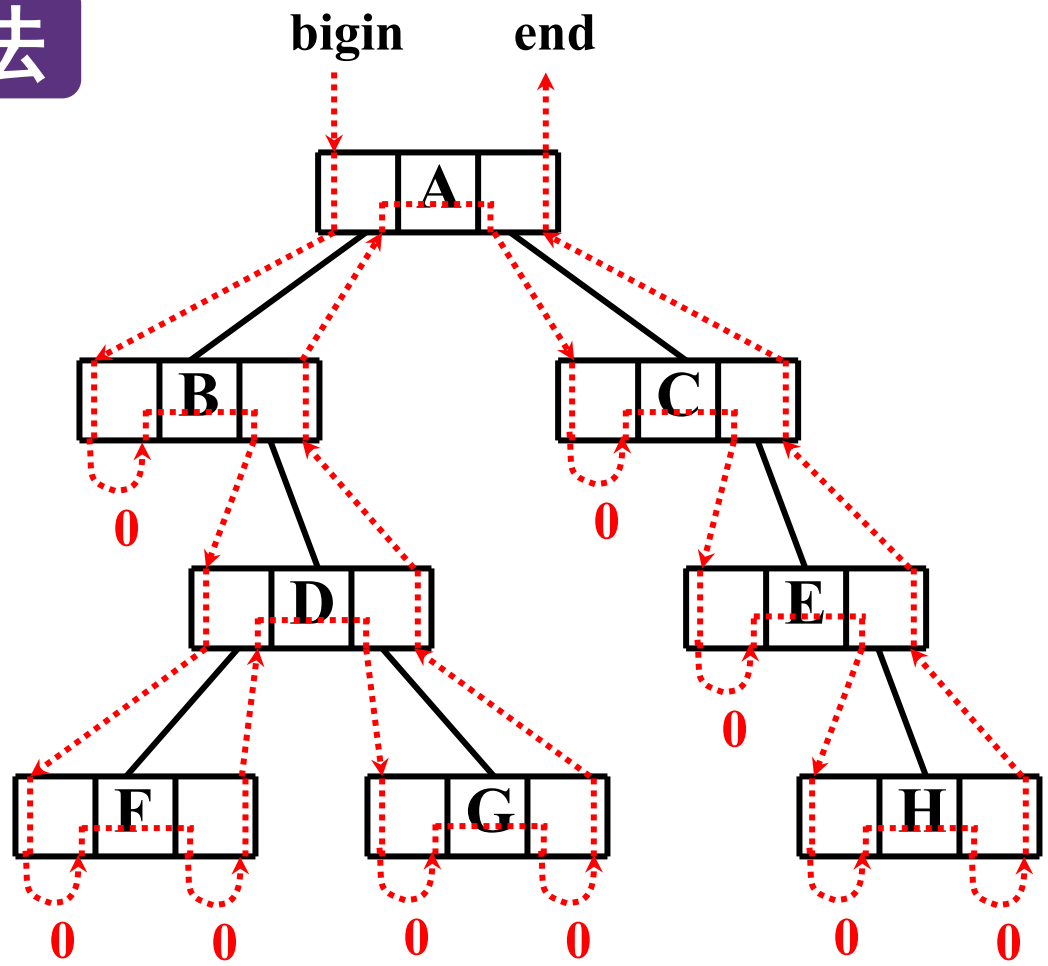
1. 中序遍历非递归方法

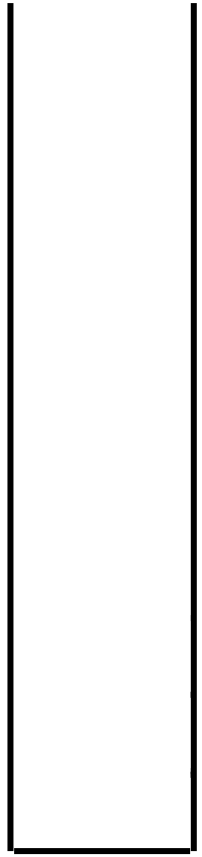
基本思想:

1、从二叉树的根结点开始，沿左枝一走到没有左孩子的结点为止，同时将所遇结点入栈，

2、待到遍历完左子树时，从栈顶退出结点并访问，然后再遍历其右子树.

3、重复上述操作完成遍历。





中序遍历顺序: B F D G A C E H

5.8 扩展与提高

5-8-2　线索二叉树

5.8.2 线索二叉树

1. 线索二叉树

指向线性序列中“前驱”和“后继”的指针称为“线索”。

包含“线索”的存储结构称为“线索链表”。

与“线索链表”相对应的二叉树称为“线索二叉树”。

在二叉链表的基础上，给每个结点加入两个标志。

lchild	ltag	data	rtag	rchild

ltag = { 0 lchild域指向结点的左子树（左子树不空）；1 lchild域指向结点的前驱结点（左子树空）

rtag = { 0 rchild域指向结点的右子树（右子树不空）；1 rchild域指向结点的后继结点（右子树空）

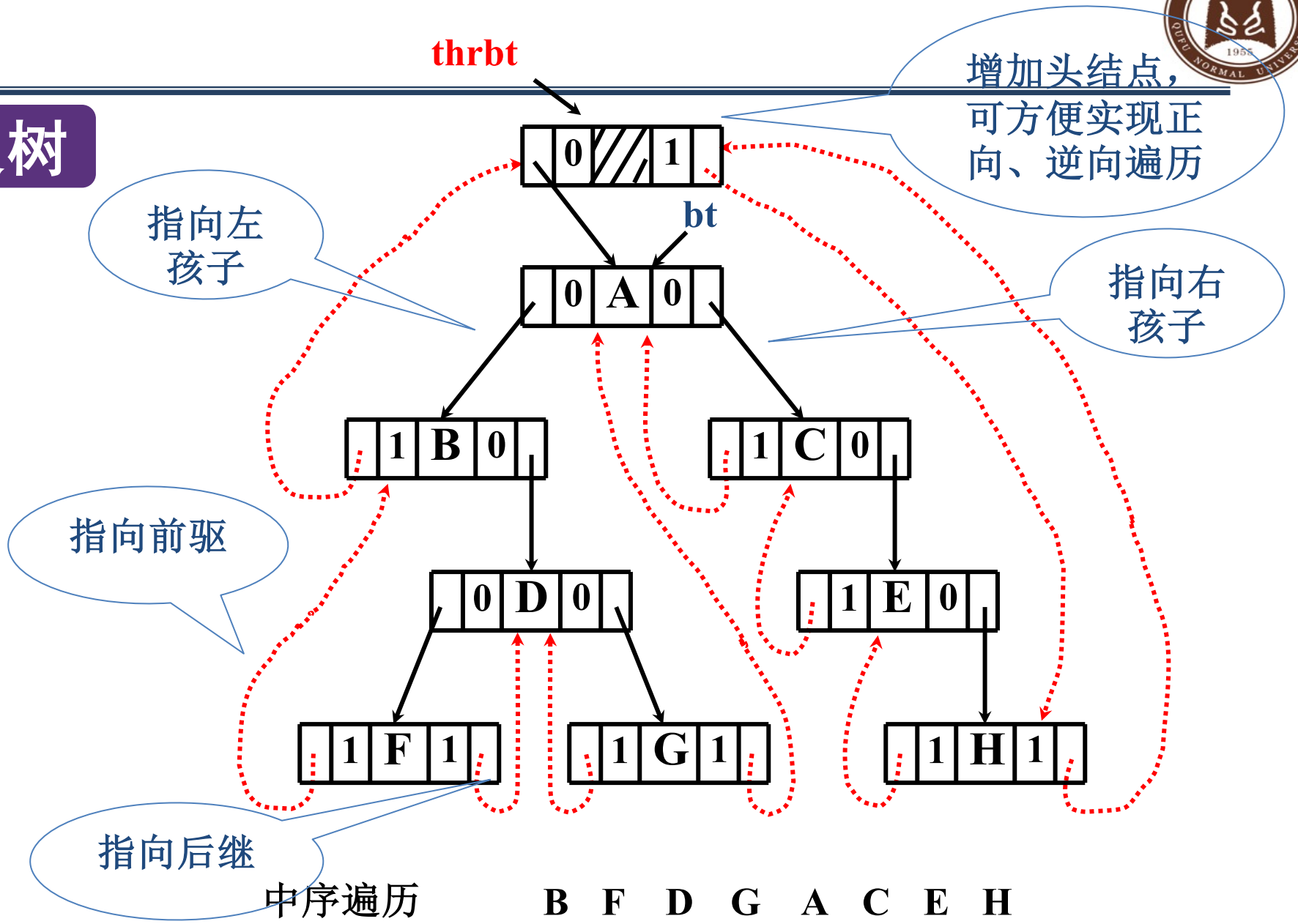
```
typedef  struct  ThrNode {
    DataType      data ;
    ThrNode    * lchild , * rchild ;
    int   ltag , rtag ;  //左右标志
}
```

5.8.2 线索二叉树

2. 中序线索二叉树

ltag = { 0 左子树；1 前驱

rtag = { 0 右子树；1 后继

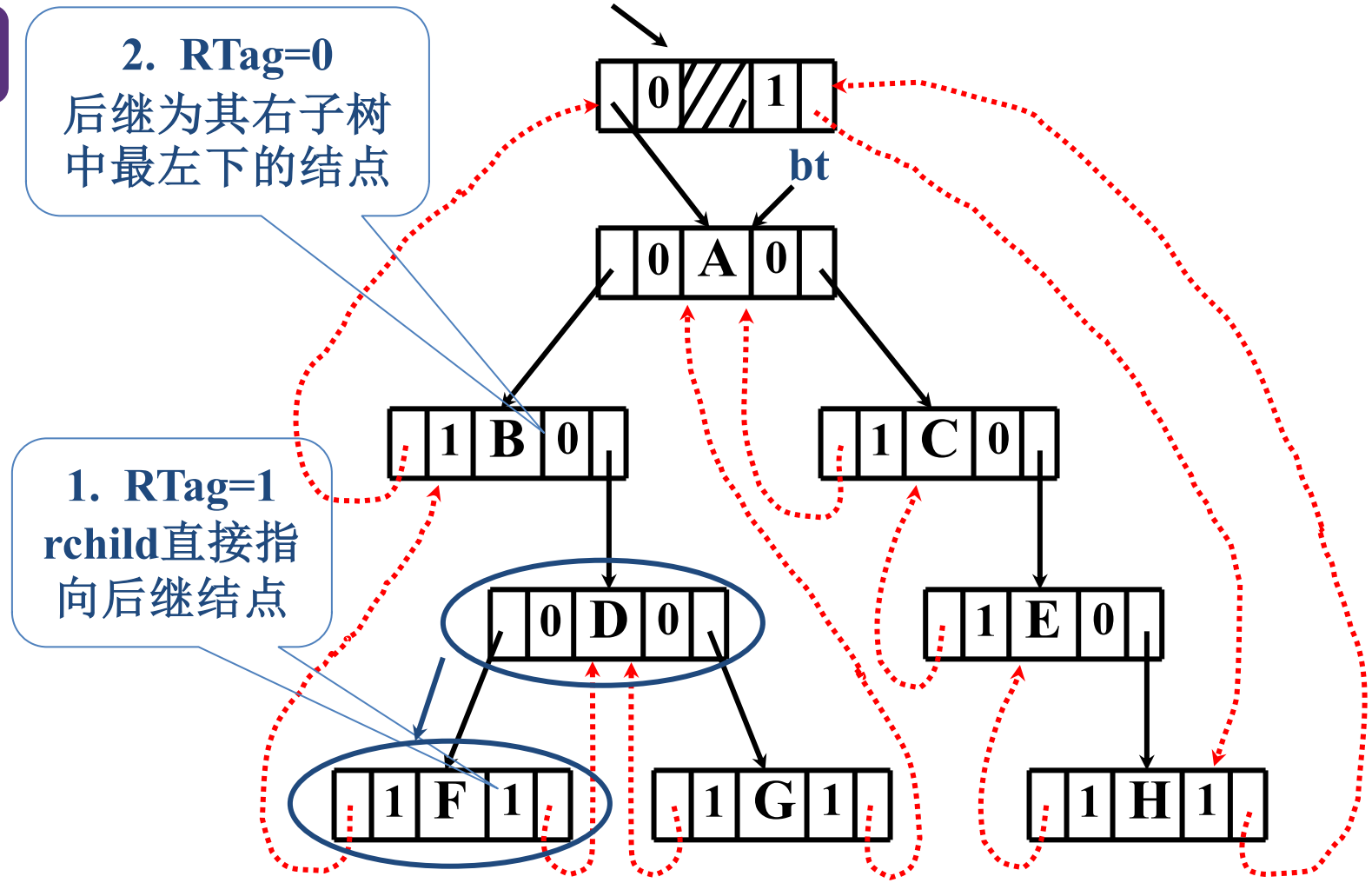


5.8.2 线索二叉树

2. 中序线索二叉树

$$ltag = \begin{cases} 0 & \text{左子树} \\ 1 & \text{前驱} \end{cases}$$

$$rtag = \begin{cases} 0 & \text{右子树} \\ 1 & \text{后继} \end{cases}$$



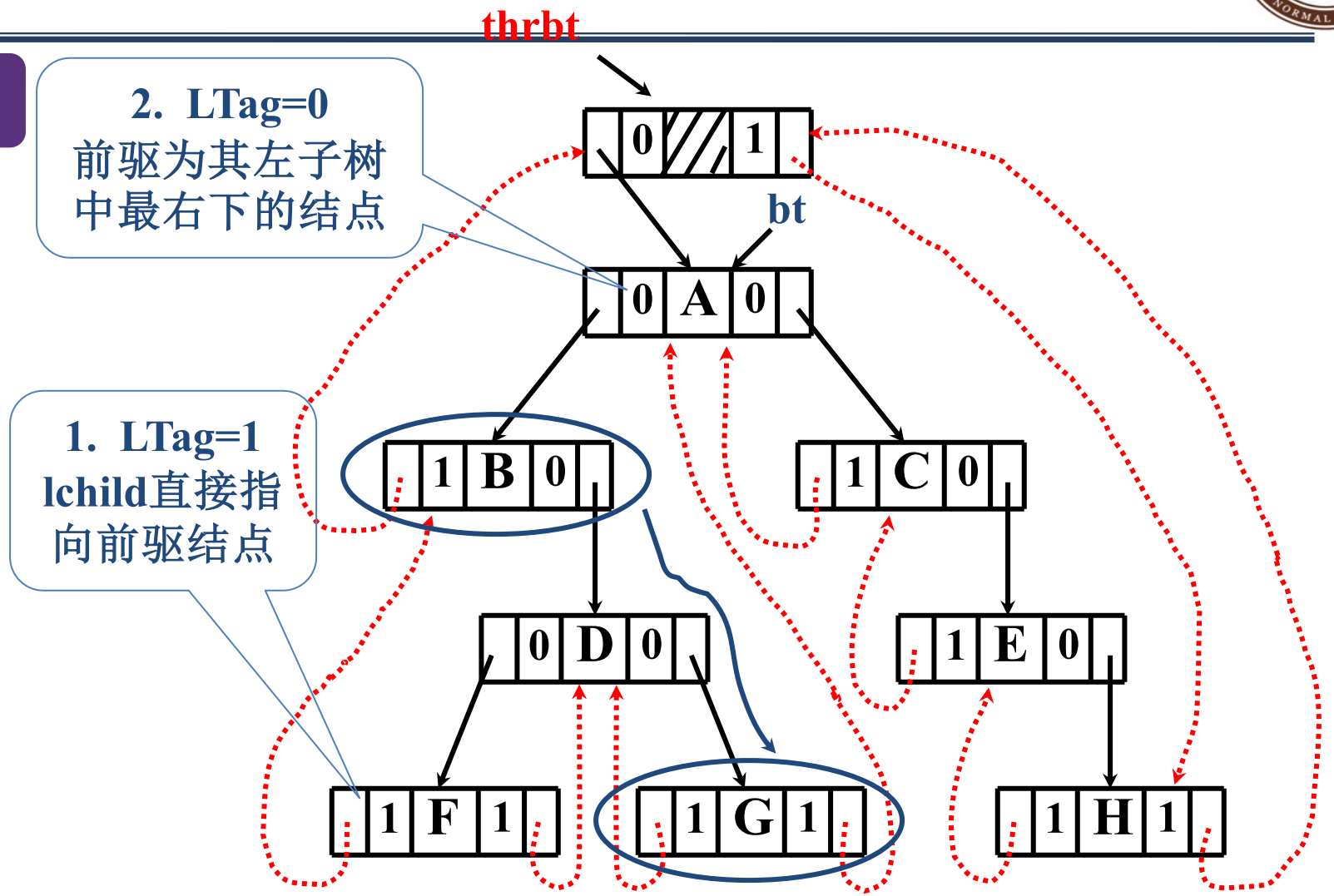
中序遍历　　B F D G A C E H

5.8.2 线索二叉树

2. 中序线索二叉树

$$ltag = \begin{cases} 0 & \text{左子树} \\ 1 & \text{前驱} \end{cases}$$

$$rtag = \begin{cases} 0 & \text{右子树} \\ 1 & \text{后继} \end{cases}$$



中序遍历　　B F D G A C E H

小结

1. 了解中序遍历二叉树的非递归算法
2. 了解线索二叉树

本章小结

1. 掌握树的定义（递归）、ADT定义及遍历方法
2. 掌握树的存储结构（双亲表示/孩子表示/孩子兄弟表示）
3. 熟练掌握二叉树的递归定义和基本性质
4. 理解二叉树的抽象数据类型定义
5. 熟练掌握二叉树的遍历方法（前序、后序、中序、层次）
6. 理解二叉树的顺序存储结构及其特点
7. 熟练掌握二叉链表的定义及节点结构
8. 熟练掌握二叉树遍历的递归方法
9. 掌握二叉树的层序遍历方法
10. 掌握二叉树的建立和销毁方法
11. 掌握树、森林和二叉树之间的转换方法
12. 掌握建立最优二叉树和哈夫曼编码的方法

Thank You !

Q & A