

本节内容

# 二叉树

## 先/中/后序 遍历

王道考研/CSKAOYAN.COM

1

知识总览

二叉树的遍历

- 先序遍历
- 中序遍历
- 后序遍历
- 遍历算法的应用举例

王道考研/CSKAOYAN.COM

2

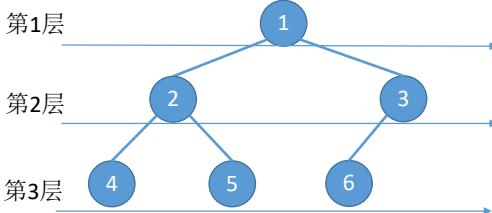
## 什么是遍历



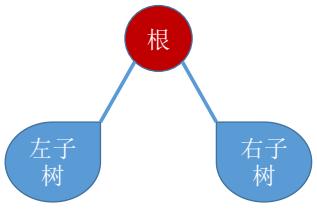
遍历：按照某种次序把所有结点都访问一遍


线性结构

简单来说



层次遍历：基于树的层次特性确定的次序规则



先/中/后序遍历：基于树的递归特性确定的次序规则

王道考研/CSKAOYAN.COM

3

## 二叉树的遍历

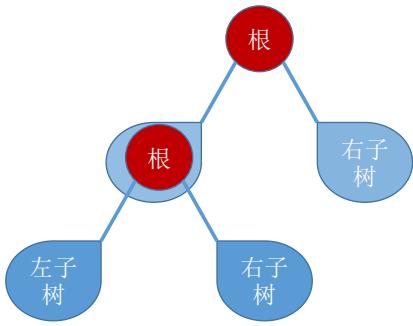


二叉树的递归特性：

- ①要么是个空二叉树
- ②要么就是由“根节点+左子树+右子树”组成的二叉树



空二叉树



非空二叉树

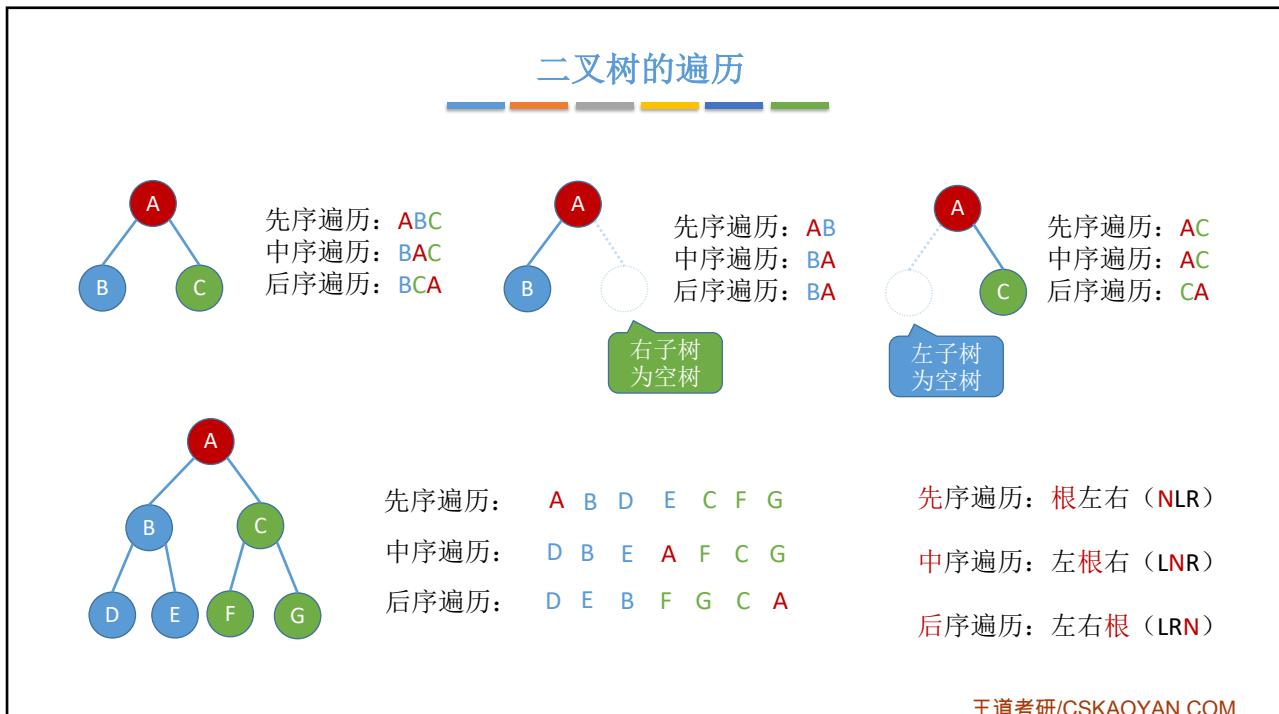
先序遍历：根左右 (NLR)

中序遍历：左根右 (LNR)

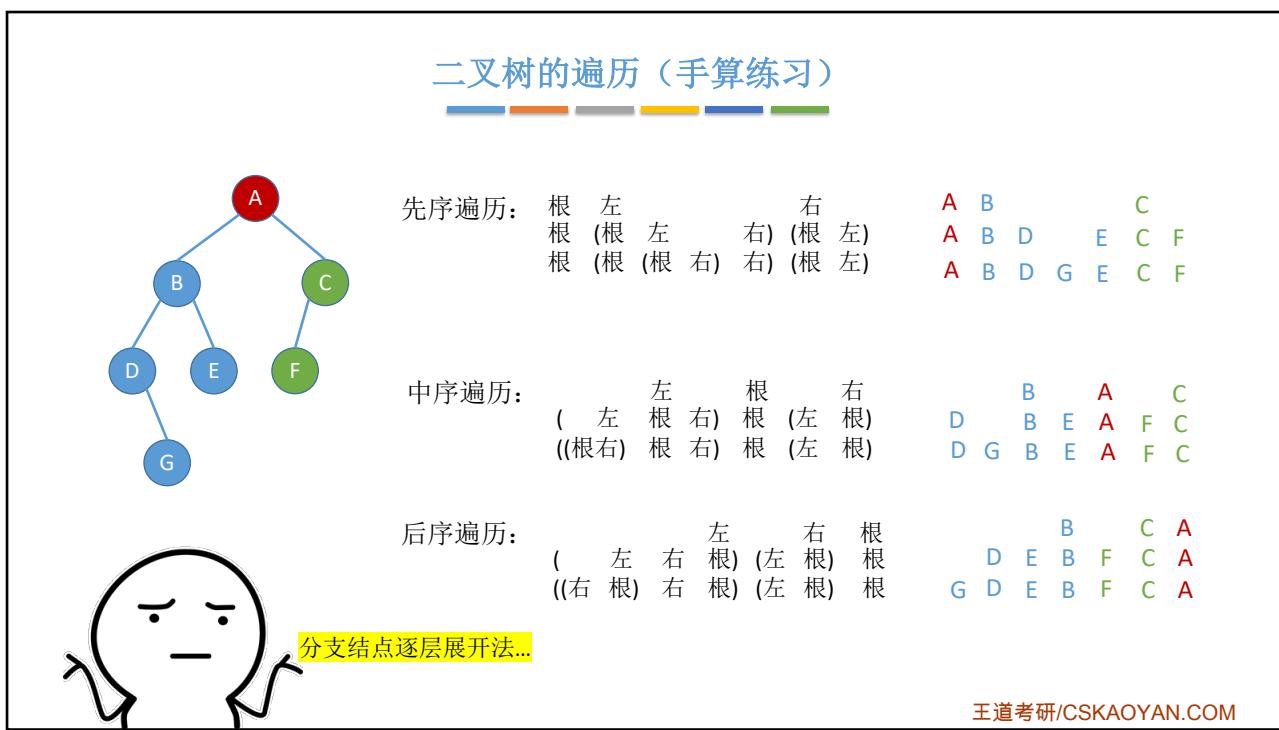
后序遍历：左右根 (LRN)

王道考研/CSKAOYAN.COM

4



5



6

### 二叉树的遍历（手算练习）

先序遍历: -+a\*b-cd/ef  
 中序遍历: a+b\*c-d-e/f  
 后序遍历: abcd-\*+ef/-

先序遍历 → 前缀表达式  
 中序遍历 → 中缀表达式 (需要加界限符)  
 后序遍历 → 后缀表达式

算数表达式的“分析树”  
 $a + b * (c - d) - e / f$

王道考研/CSKAOYAN.COM

7

### 先序遍历（代码）

先序遍历 (PreOrder) 的操作过程如下：  
 1. 若二叉树为空，则什么也不做；  
 2. 若二叉树非空：  
 ①访问根结点；  
 ②先序遍历左子树；  
 ③先序遍历右子树。

```

//先序遍历
void PreOrder(BiTree T){
    if(T!=NULL){
        visit(T); //访问根结点
        PreOrder(T->lchild); //递归遍历左子树
        PreOrder(T->rchild); //递归遍历右子树
    }
}

```

王道考研/CSKAOYAN.COM

8

### 中序遍历（代码）

中序遍历（InOrder）的操作过程如下：

1. 若二叉树为空，则什么也不做；
2. 若二叉树非空：
  - ①先序遍历左子树；
  - ②访问根结点；
  - ③先序遍历右子树。

```

//中序遍历
void InOrder(BiTree T){
    if(T!=NULL){
        InOrder(T->lchild);           //递归遍历左子树
        visit(T);                      //访问根结点
        InOrder(T->rchild);           //递归遍历右子树
    }
}

```

王道考研/CSKAOYAN.COM

9

### 后序遍历（代码）

后序遍历（InOrder）的操作过程如下：

1. 若二叉树为空，则什么也不做；
2. 若二叉树非空：
  - ①先序遍历左子树；
  - ②先序遍历右子树；
  - ③访问根结点。

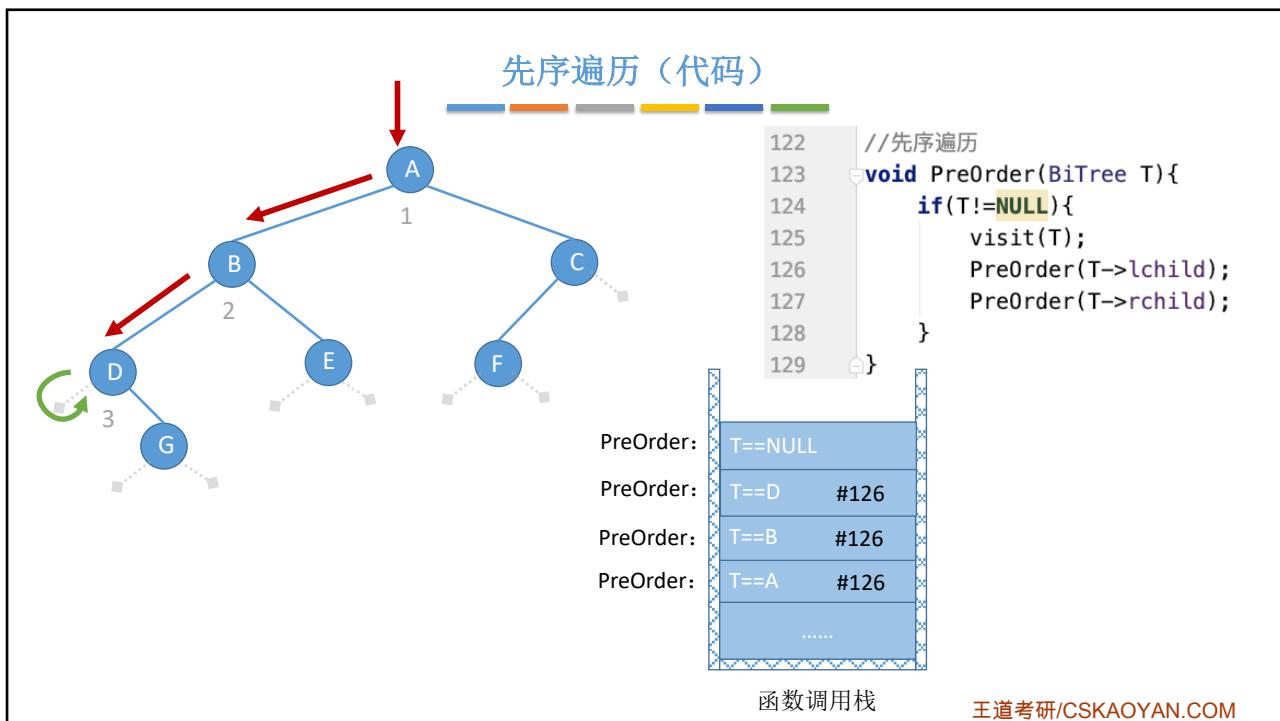
```

//后序遍历
void PostOrder(BiTree T){
    if(T!=NULL){
        PostOrder(T->lchild);        //递归遍历左子树
        PostOrder(T->rchild);        //递归遍历右子树
        visit(T);                      //访问根结点
    }
}

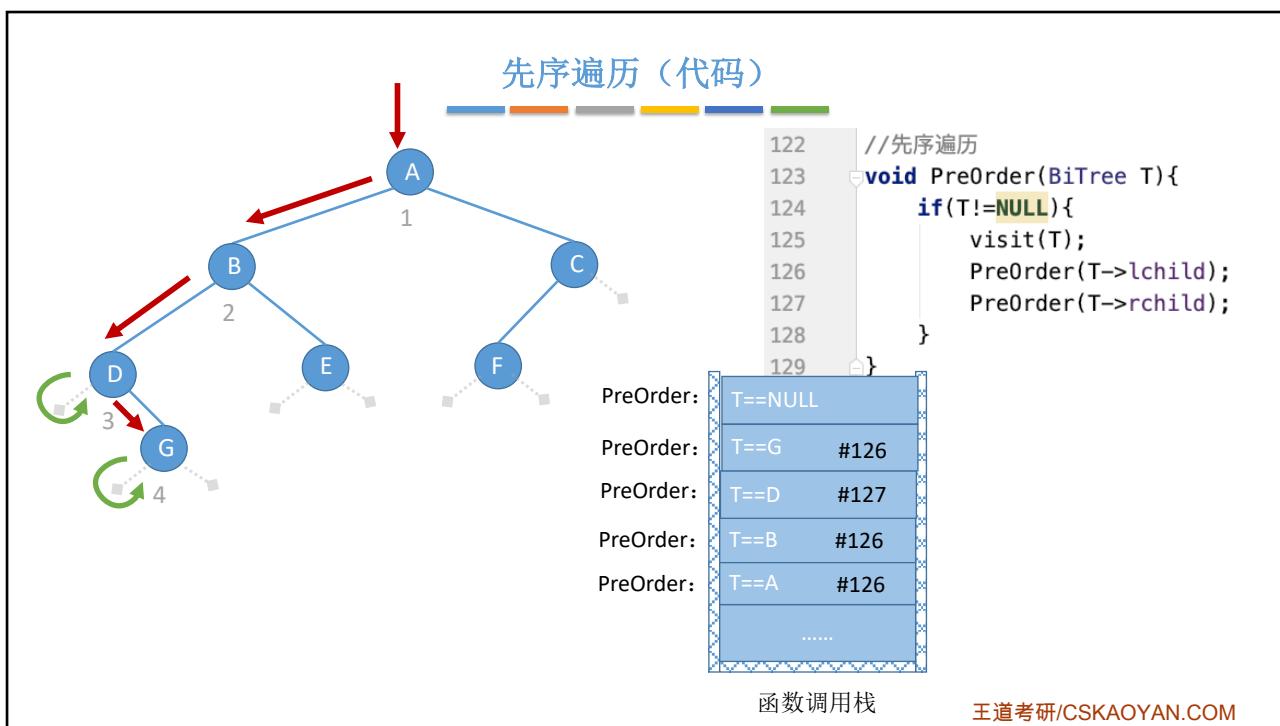
```

王道考研/CSKAOYAN.COM

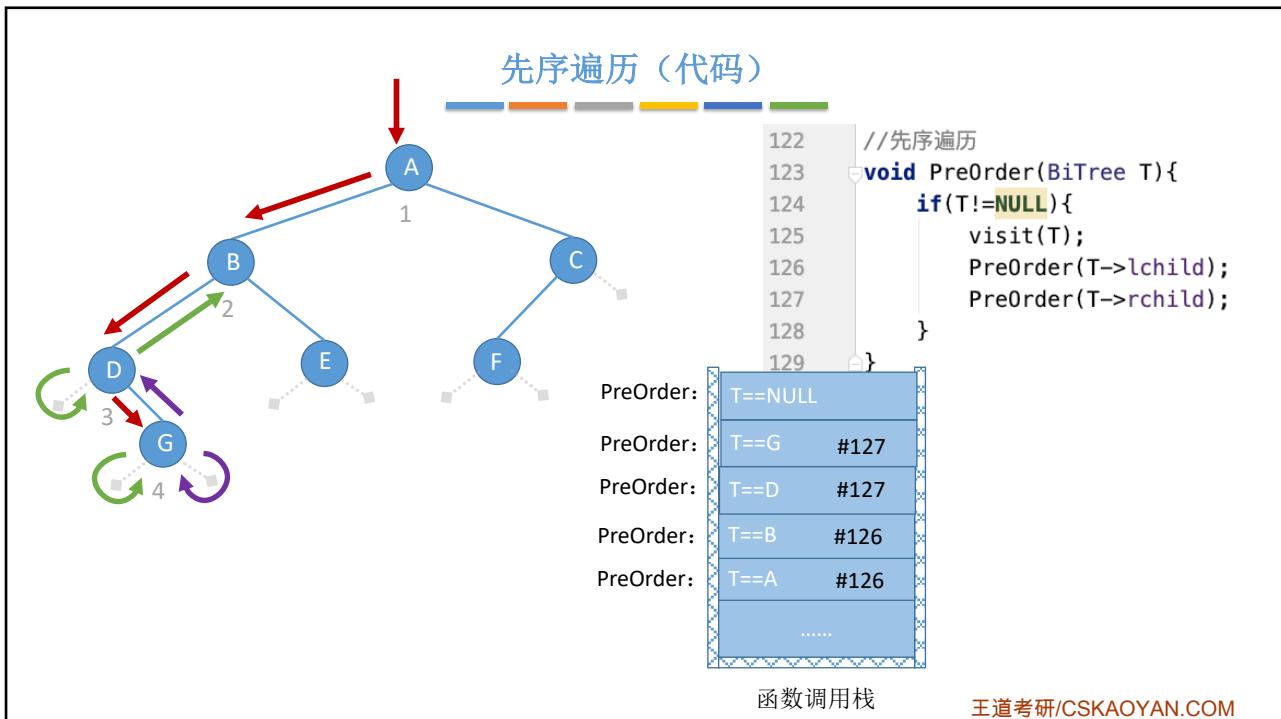
10



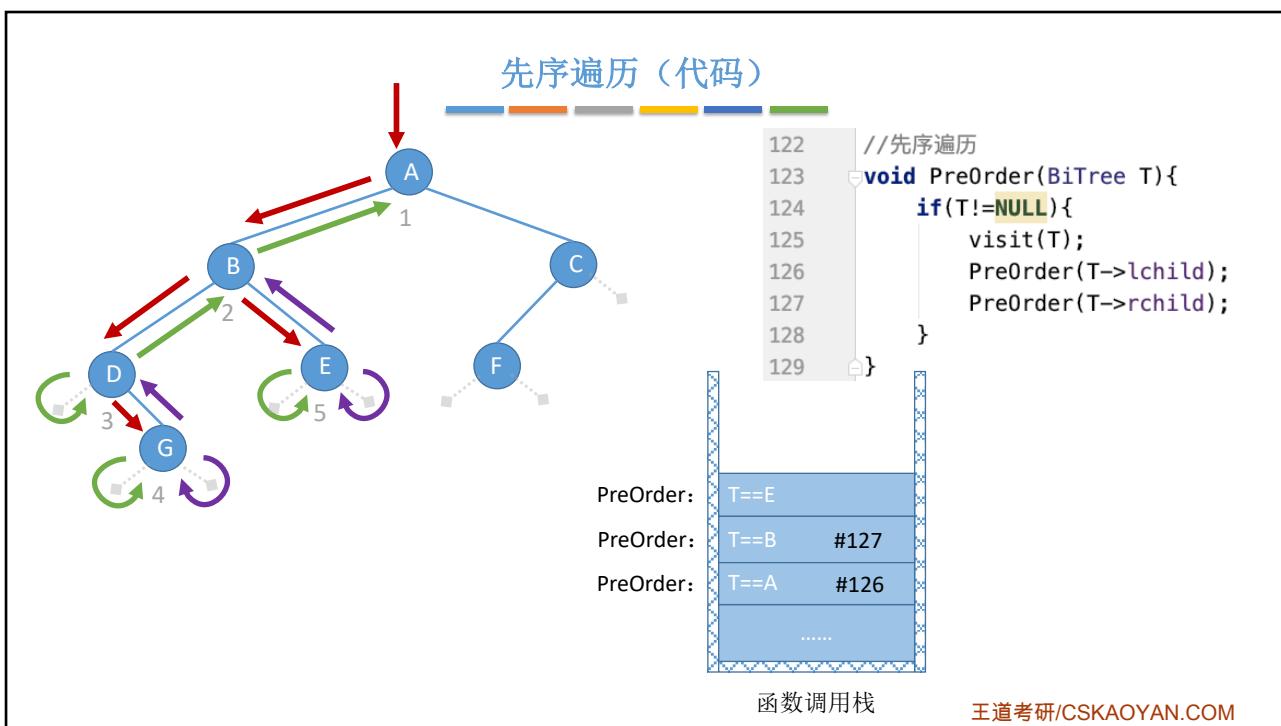
11



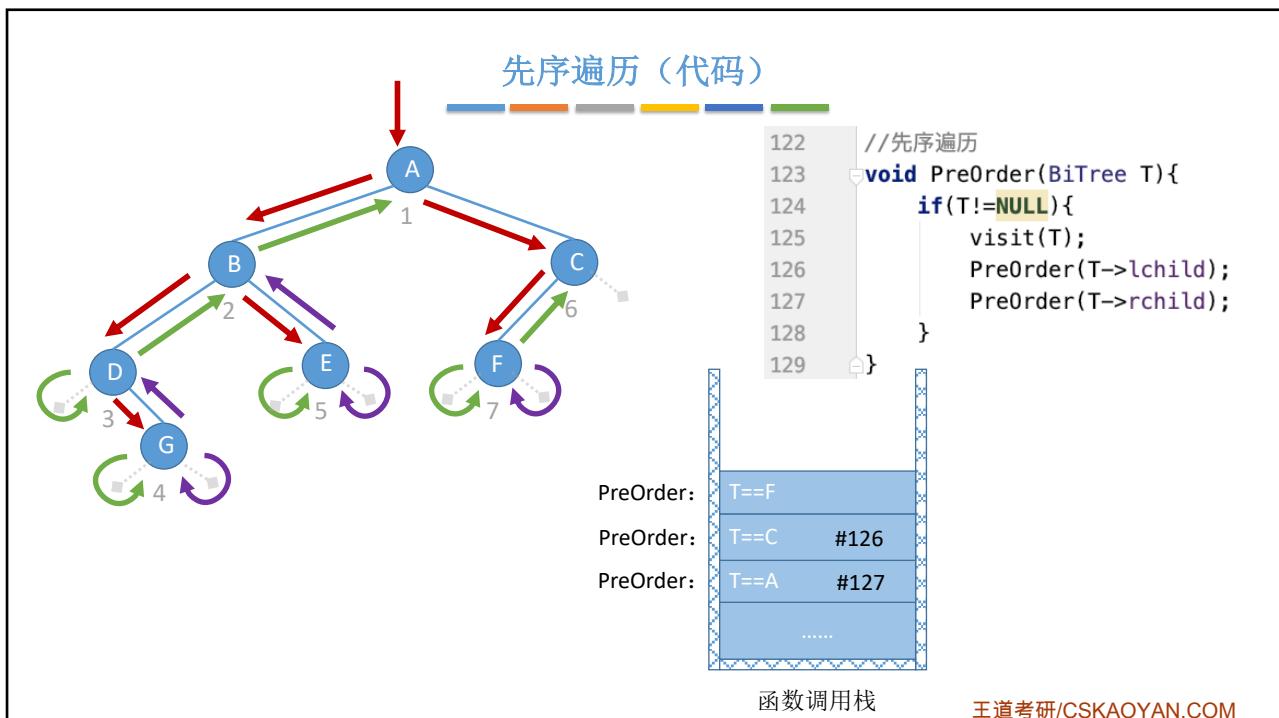
12



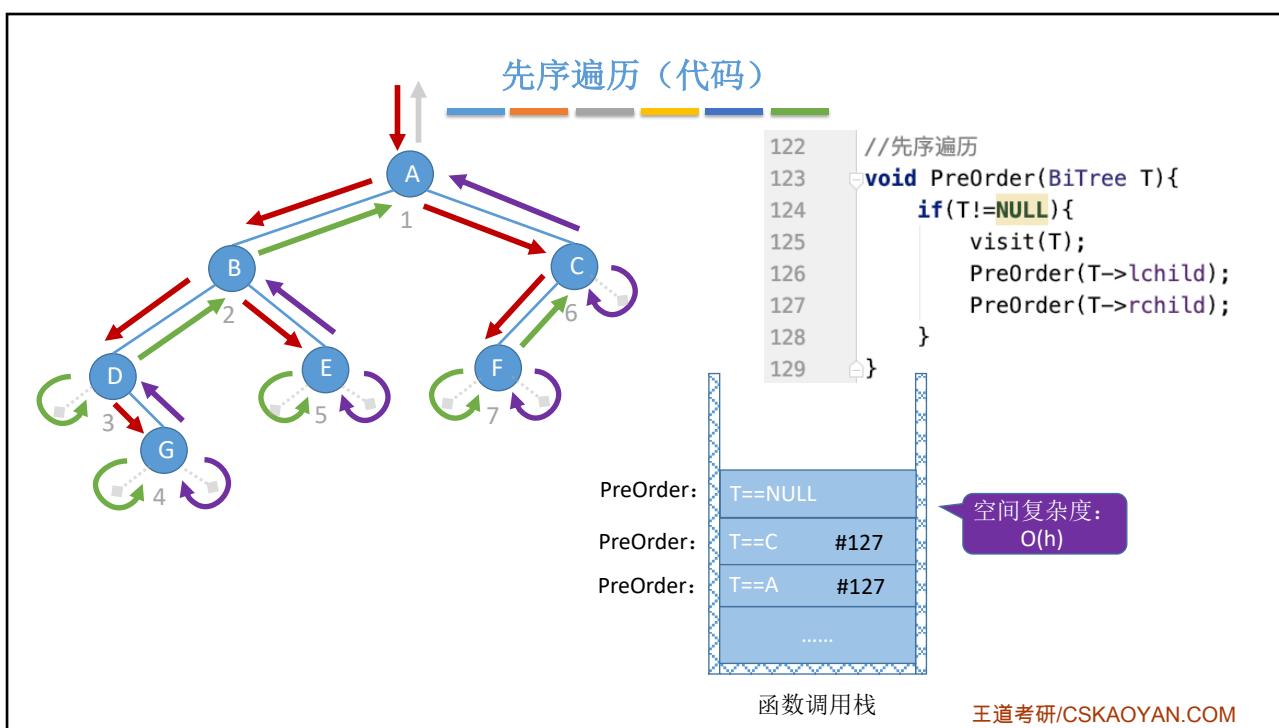
13



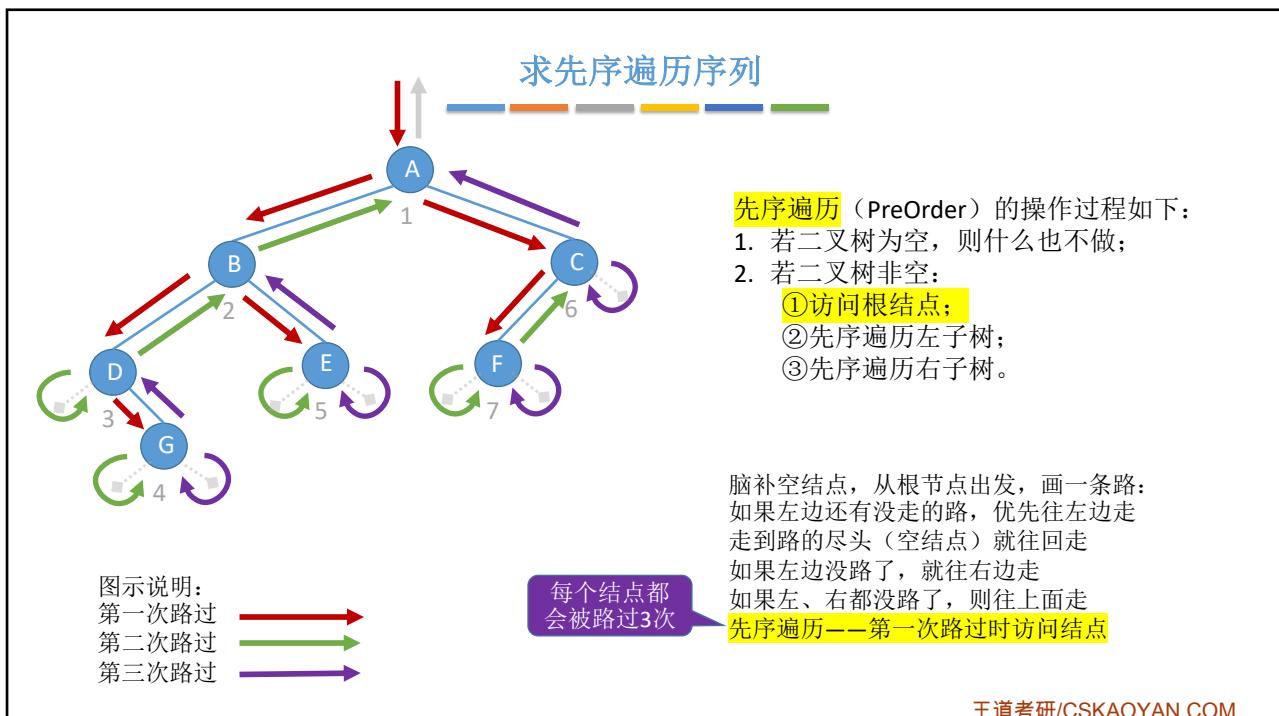
14



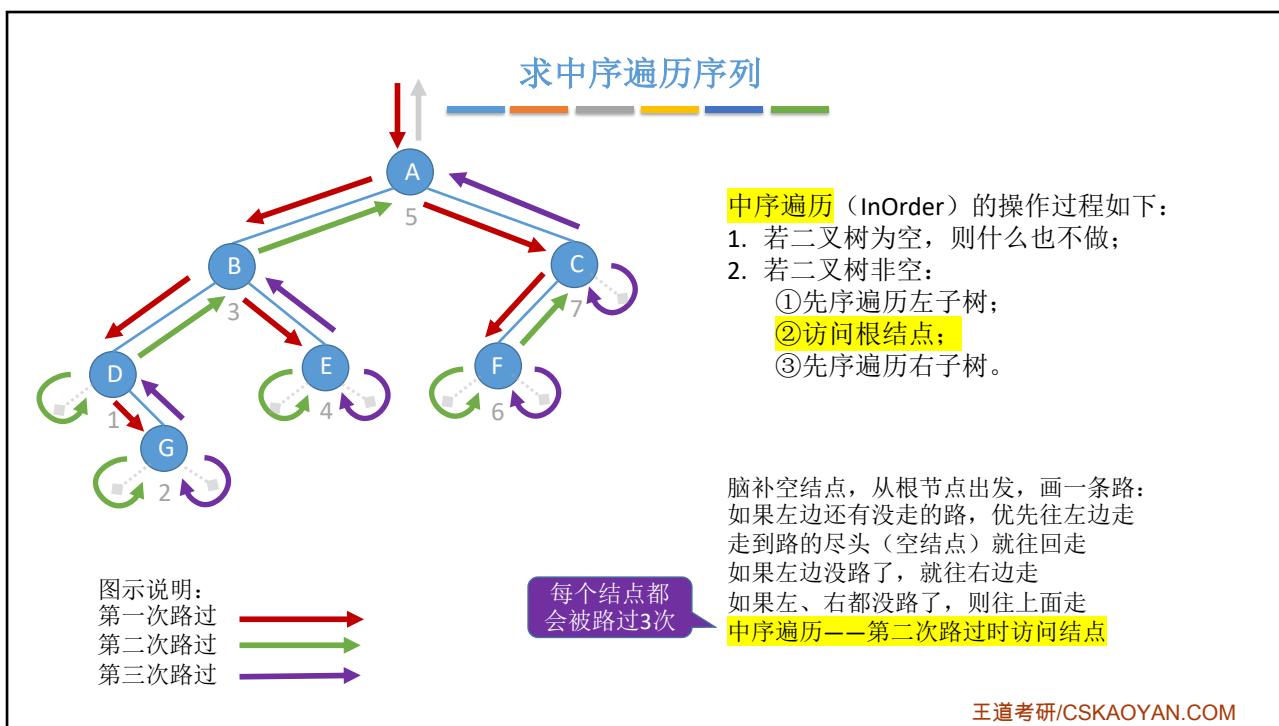
15



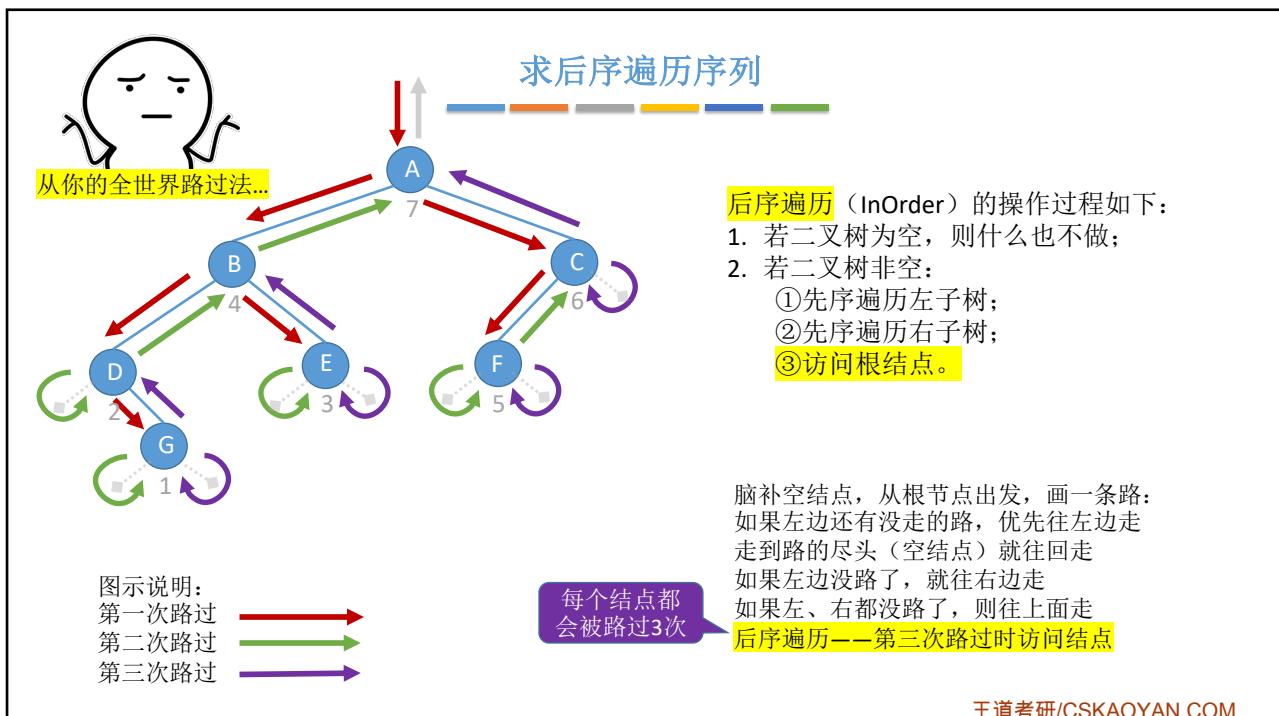
16



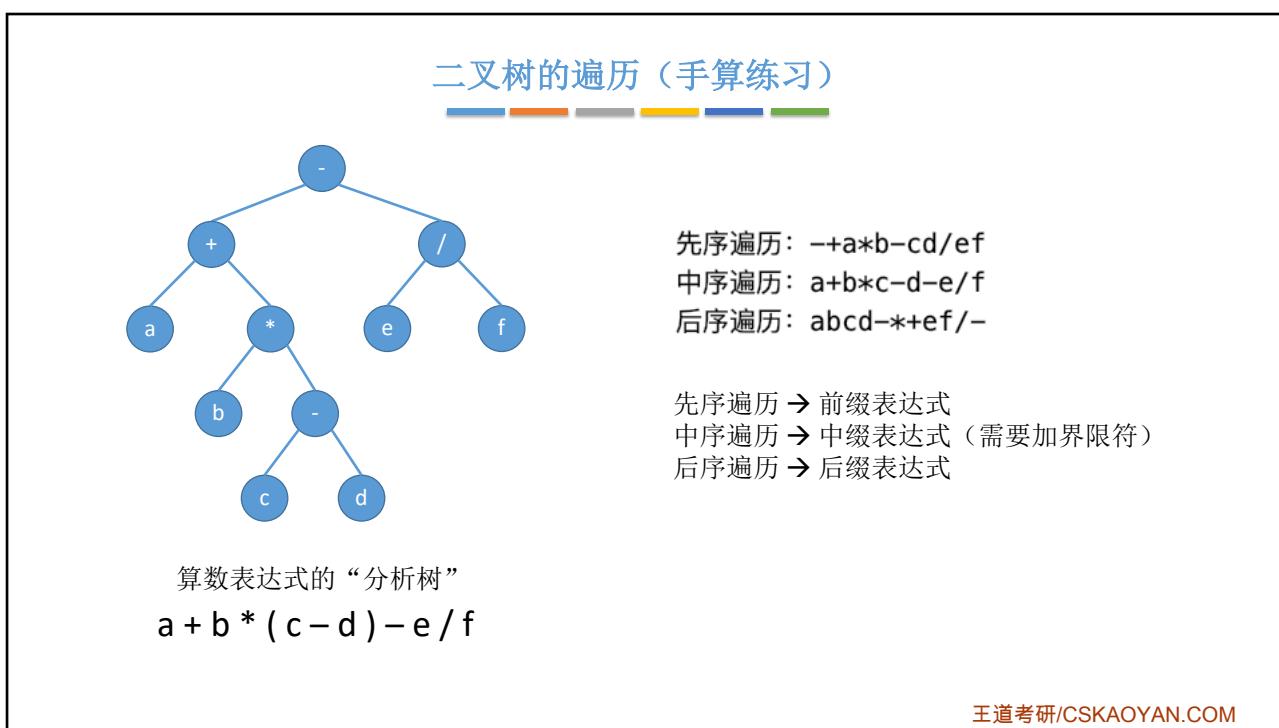
17



18



19



20

例：求树的深度（应用）

```

int treeDepth(BiTree T){
    if (T == NULL) {
        return 0;
    }
    else {
        int l = treeDepth(T->lchild);
        int r = treeDepth(T->rchild);
        //树的深度=Max(左子树深度, 右子树深度)+1
        return l>r ? l+1 : r+1;
    }
}

```

王道考研/CSKAOYAN.COM

21

知识回顾与重要考点

空间复杂度：  
 $O(h)$

三种方法

先序遍历 根、左、右

中序遍历 左、根、右

后序遍历 左、右、根

先序遍历得前缀表达式

中序遍历得中缀表达式 (没有括号)

后序遍历得后缀表达式

分支结点逐层展开法...

从你的全世界路过法

先序——第一次路过时访问  
中序——第二次路过时访问  
后序——第三次路过时访问

王道考研/CSKAOYAN.COM

22