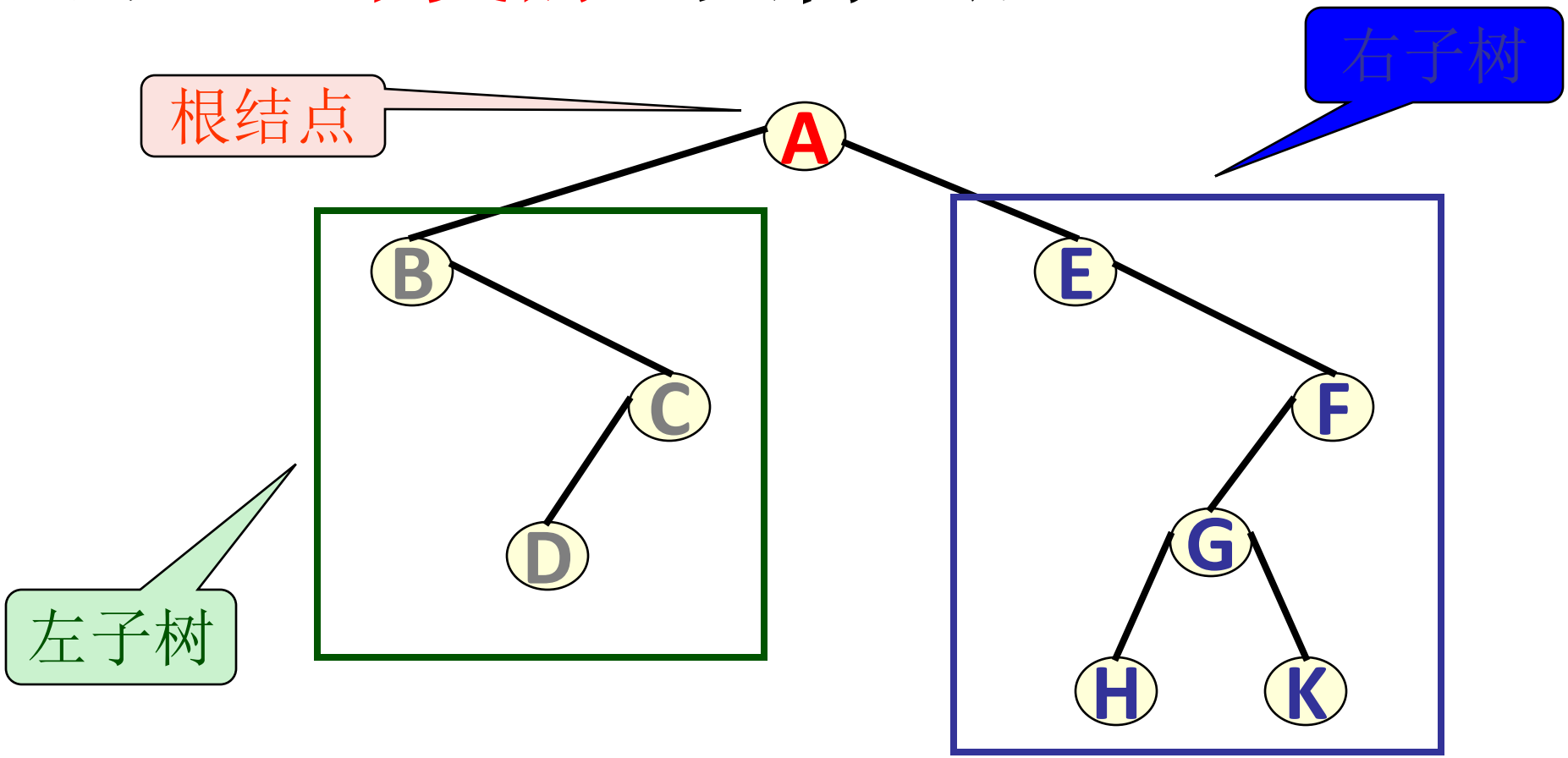
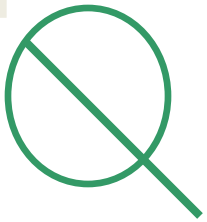
二叉树或为**空树**，或是由一个**根结点**加上**两棵**分别称为**左子树**和右子树的、**互不交的**二叉树组成。



二叉树的五种基本形态：

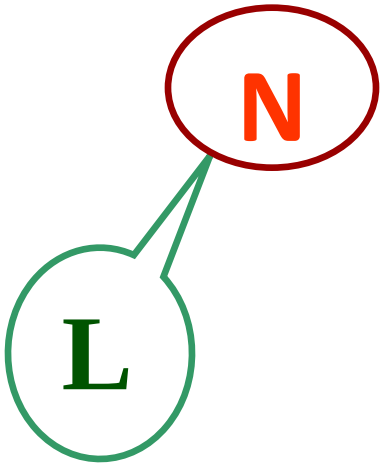
空树



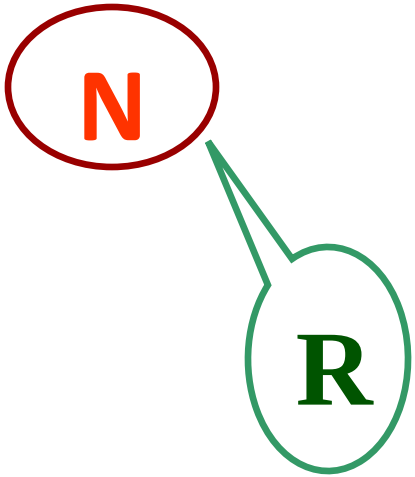
只含根结点

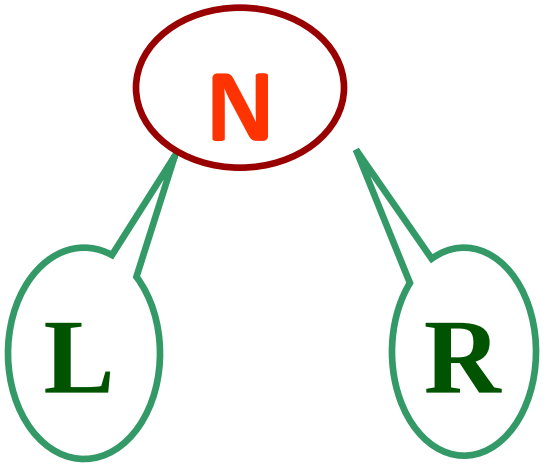
左右子树均不为空树

右子树为空树



左子树为空树





Binary Trees
（二叉树）

- Definitions:
  - A *binary tree* is either empty, or it consists of a node（结点） called the *root*（根） together with two binary trees called the *left subtree*（左子树）and the *right subtree*（右子树） of the root.

二叉树的主要基本操作：

- 查 找 类
- 插 入 类
- 删 除 类

Root(T); Value(T, e); Parent(T, e);

LeftChild(T, e); RightChild(T, e);

LeftSibling(T, e); RightSibling(T, e);

BiTreeEmpty(T); BiTreeDepth(T);

PreOrderTraverse(T, Visit());

InOrderTraverse(T, Visit());

PostOrderTraverse(T, Visit());

LevelOrderTraverse(T, Visit());

InitBiTree(&T);

Assign(T, &e, value);

CreateBiTree(&T, definition);

InsertChild(T, p, LR, c);

ClearBiTree(&T);

DestroyBiTree(&T);

DeleteChild(T, p, LR);

解读：

(1)递归定义 (recursive definition)

(2)二叉树可以为空 (empty binary tree)

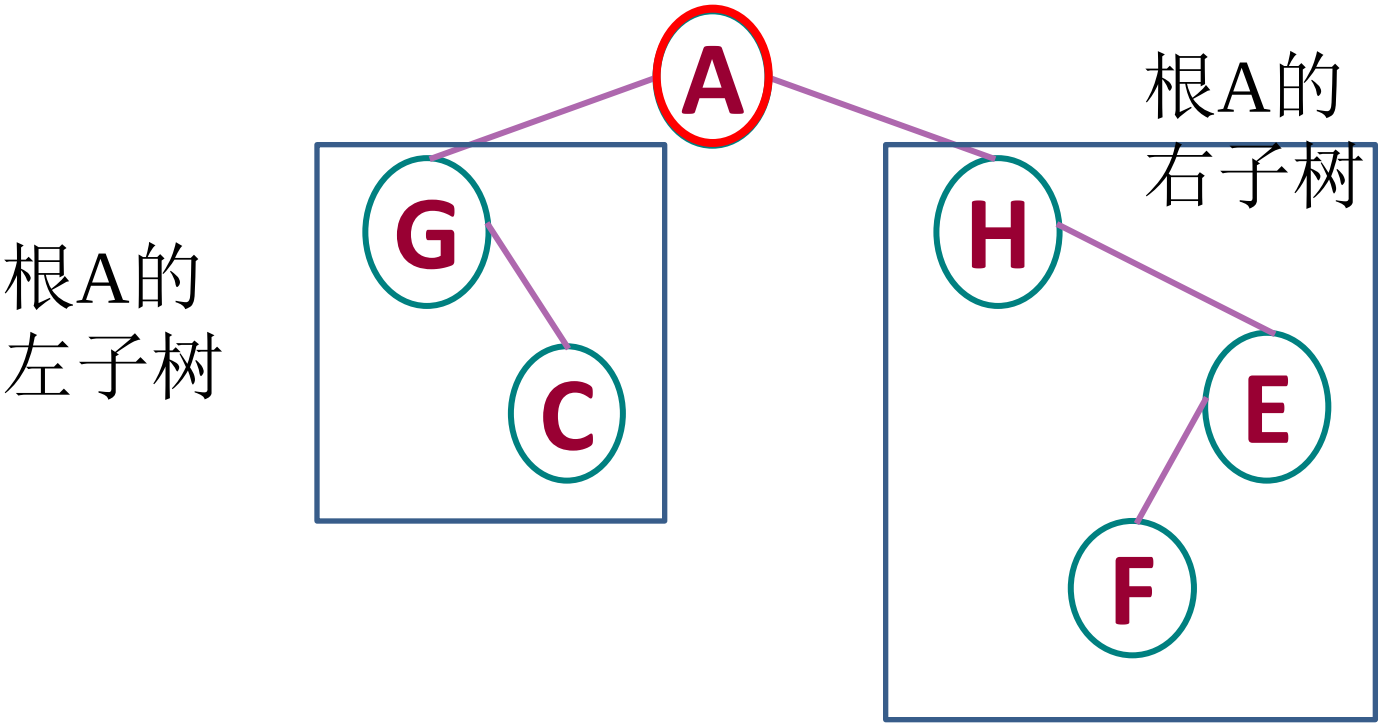
(3)结点即数据元素(Node,Data Element)

(4)非空二叉树的3个部分

a.根结点，地位尊贵，所有其它结点的祖先

b.根的左子树:一棵二叉树，空或者非空

c.根的右子树:一棵二叉树，空或者非空



(5)二叉树的子树严格区分左右

(6)结点之间的关系：

根结点与其子树的根之间的父子关系

A（双亲）-----G（左孩子）

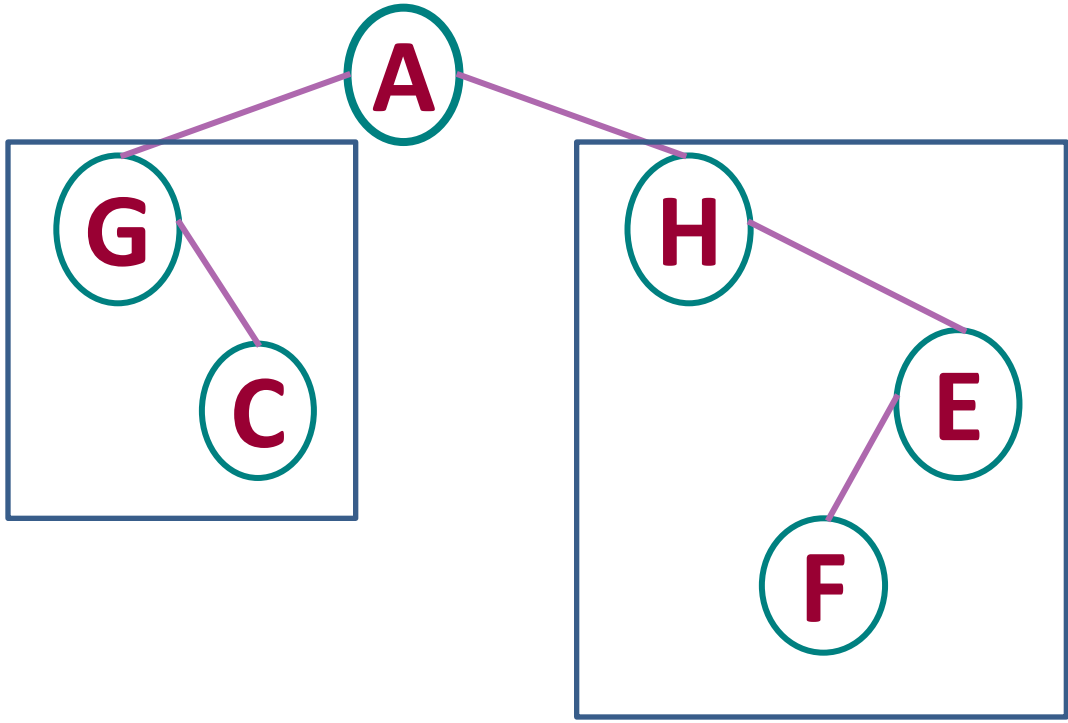
A（双亲）-----H（右孩子）

G（双亲）-----C（右孩子）

G、H 互为兄弟

H（祖先）

----E F（子孙）



(7)二叉树是一种层次型的数据结构

(8)结点的度

结点的度：结点所拥有的子树的数目。

A的度为2

E的度为1

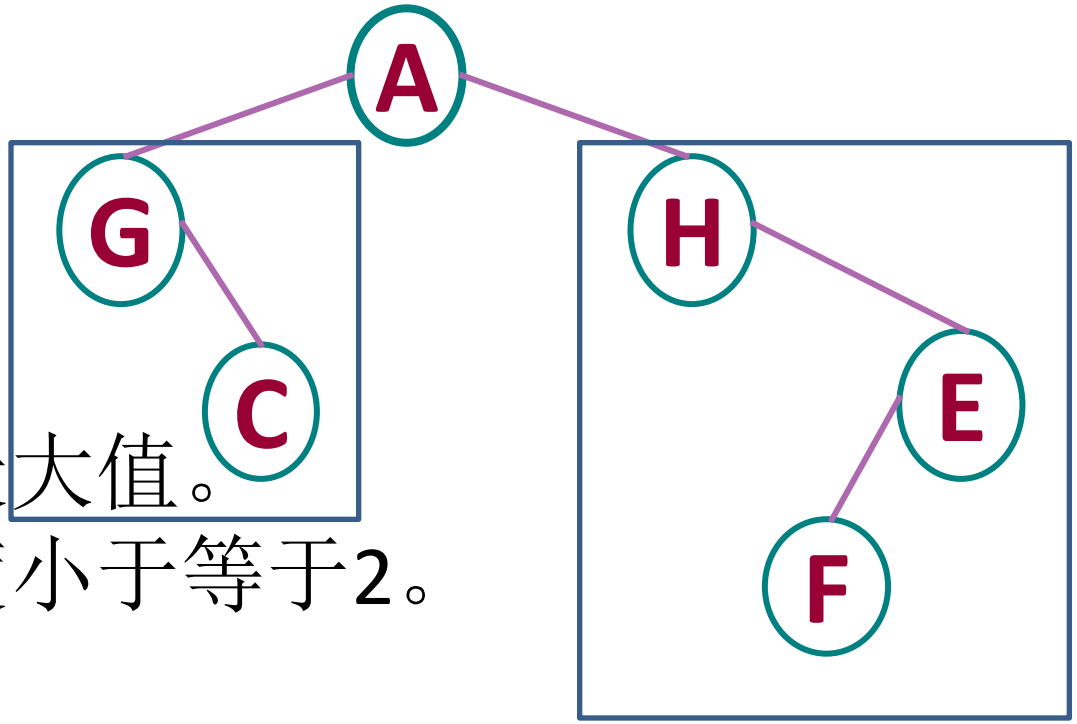
C的度为0。

叶子：度为0的结点。

(9)二叉树的度

二叉树中结点度的最大值。

任意一棵二叉树的度小于等于2。



二叉树的五种形态 Five Base Statues:

1、空二叉树
2、左右子树都不空
3、左子树不空
4、右子树不空
5、只有根结点