根深叶茂德馨，人生巍峨树立

第10讲 树和二叉树的基本概念

信息学院（智能应用研究院）
欧 新 宇

建议课时：2

第5章 树和二叉树

考核要点

- **考核大纲**

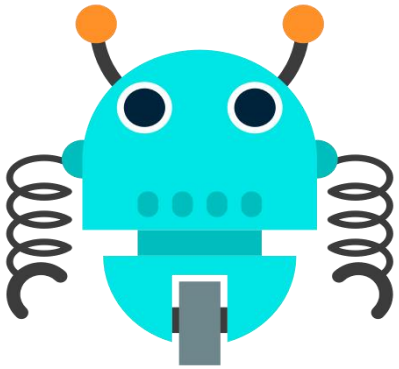
树的基本概念及存储结构；二叉树的定义、主要特征及其存储结构，二叉树的遍历，线索二叉树的基本概念和构造；森林与二叉树的的转换；树和森林的遍历；哈夫曼树和哈夫曼编码。

- **复习要点**
  - 掌握树和二叉树的性质、遍历操作、存储结构和操作特性
  - 了解满二叉树、完全二叉树、线索二叉树、哈夫曼树的定义、性质和思想
  - 熟练掌握二叉树的前、中、后序遍历方法及算法
  - 熟练掌握哈夫曼树的实现方法及构造哈夫曼编码的方法
  - 掌握森林和二叉树的转换方法

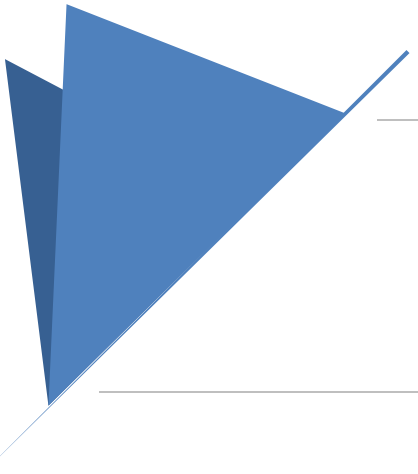
第10讲 树和二叉树的基本概念

本讲作业

- **必做题**
  - ✓ 【课堂互动10.1】 树的基本概念
  - ✓ 【课堂互动10.2】 二叉树的基本概念和性质
  - ✓ 【课堂互动10.3】 二叉树的存储结构
- **选做题**
  - ✓ 【课前自测10】树和二叉树
  - ✓ 【扩展练习10】树和二叉树



- 树的基本概念
- 二叉树的基本概念
- 二叉树的性质
- 二叉树的存储结构



树的基本概念

/ 树的示例

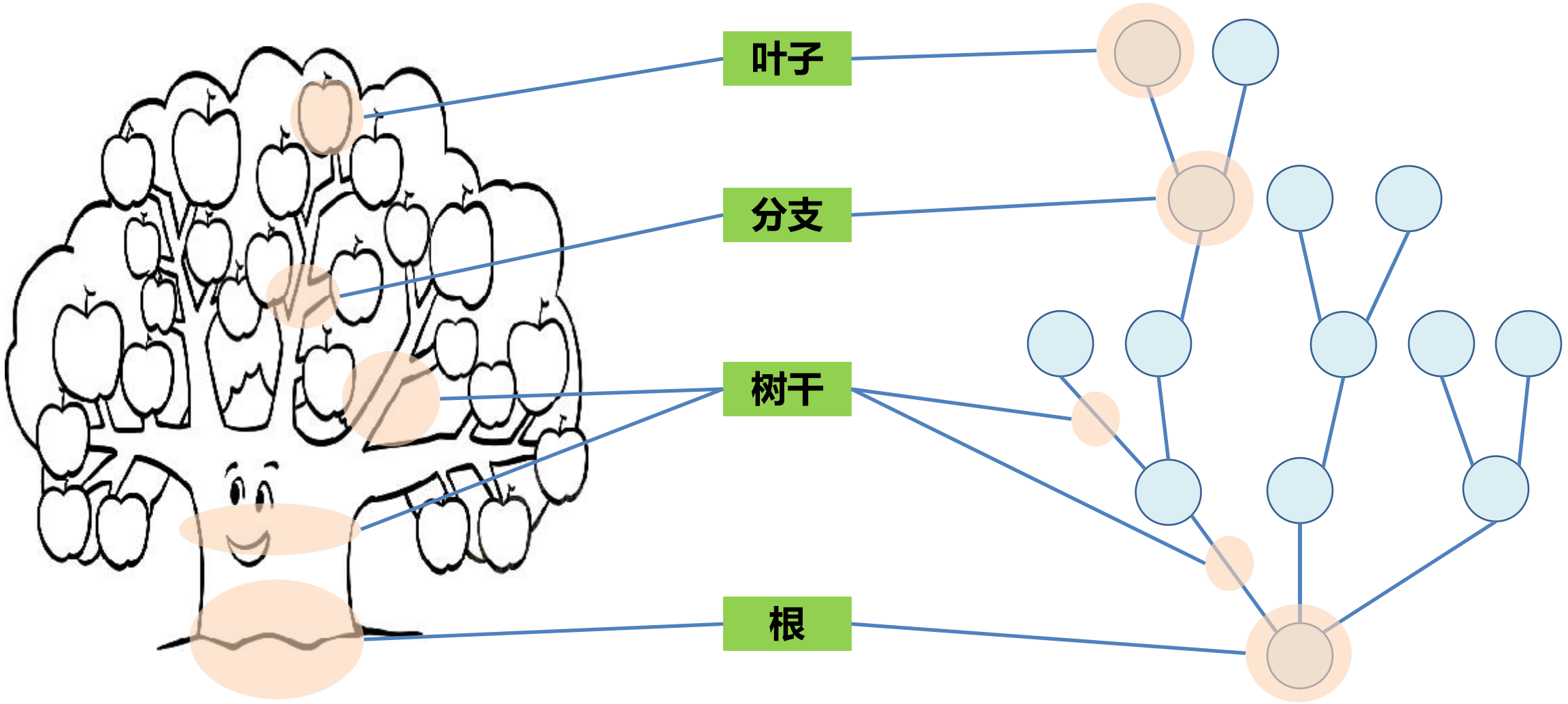
/ 树的定义

/ 树的表示方法

/ 树的基本术语

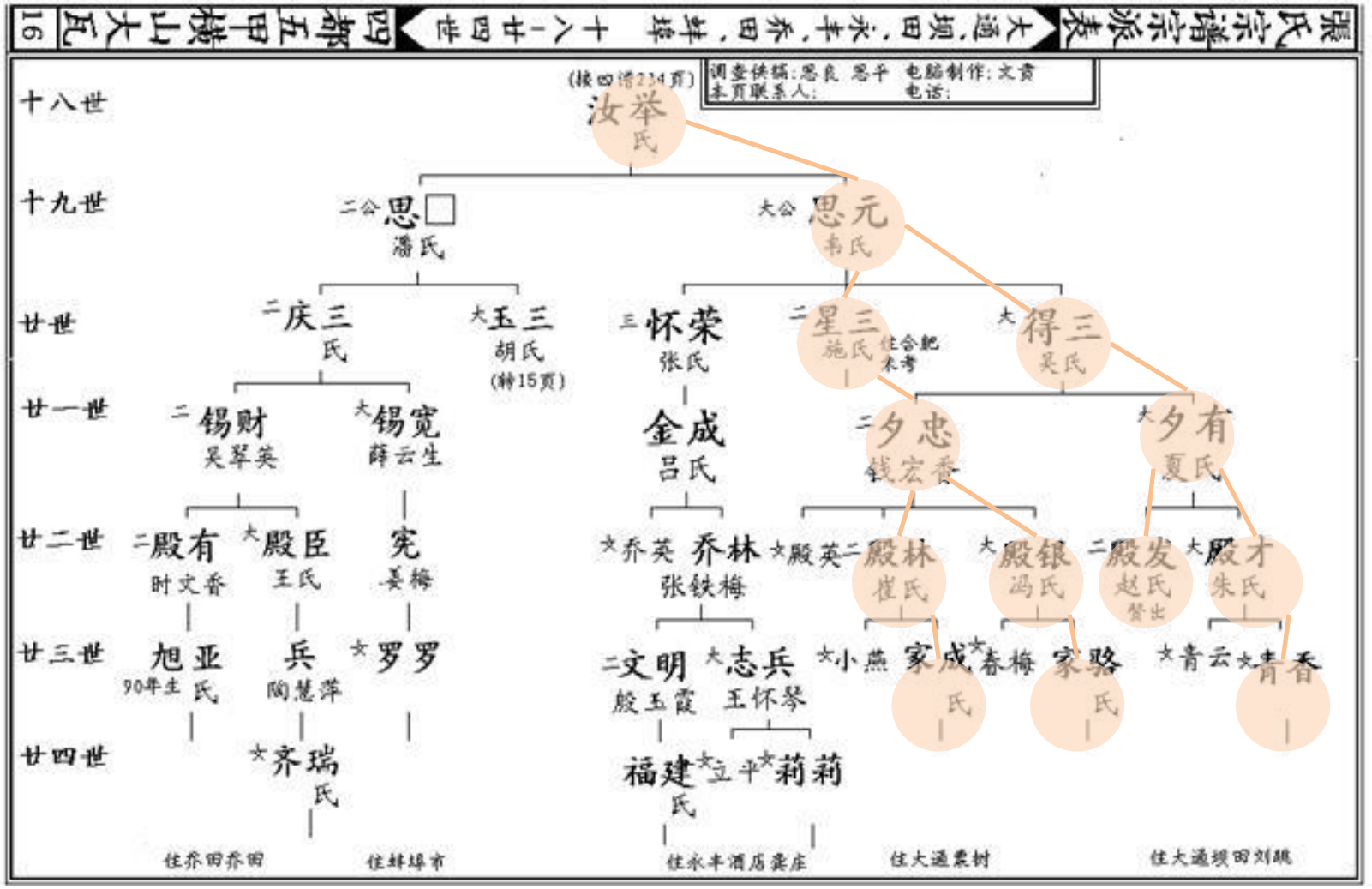
树的基本概念

什么是树



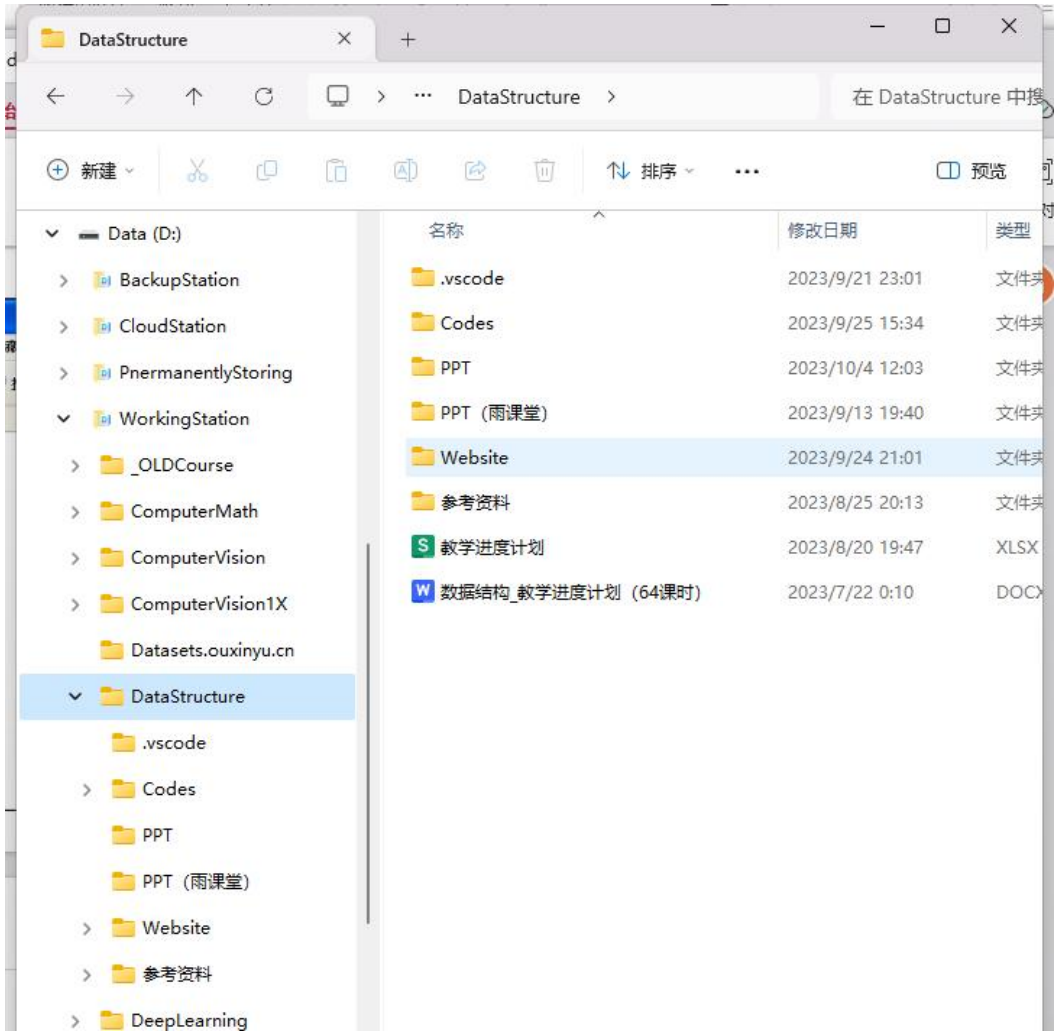
树的基本概念

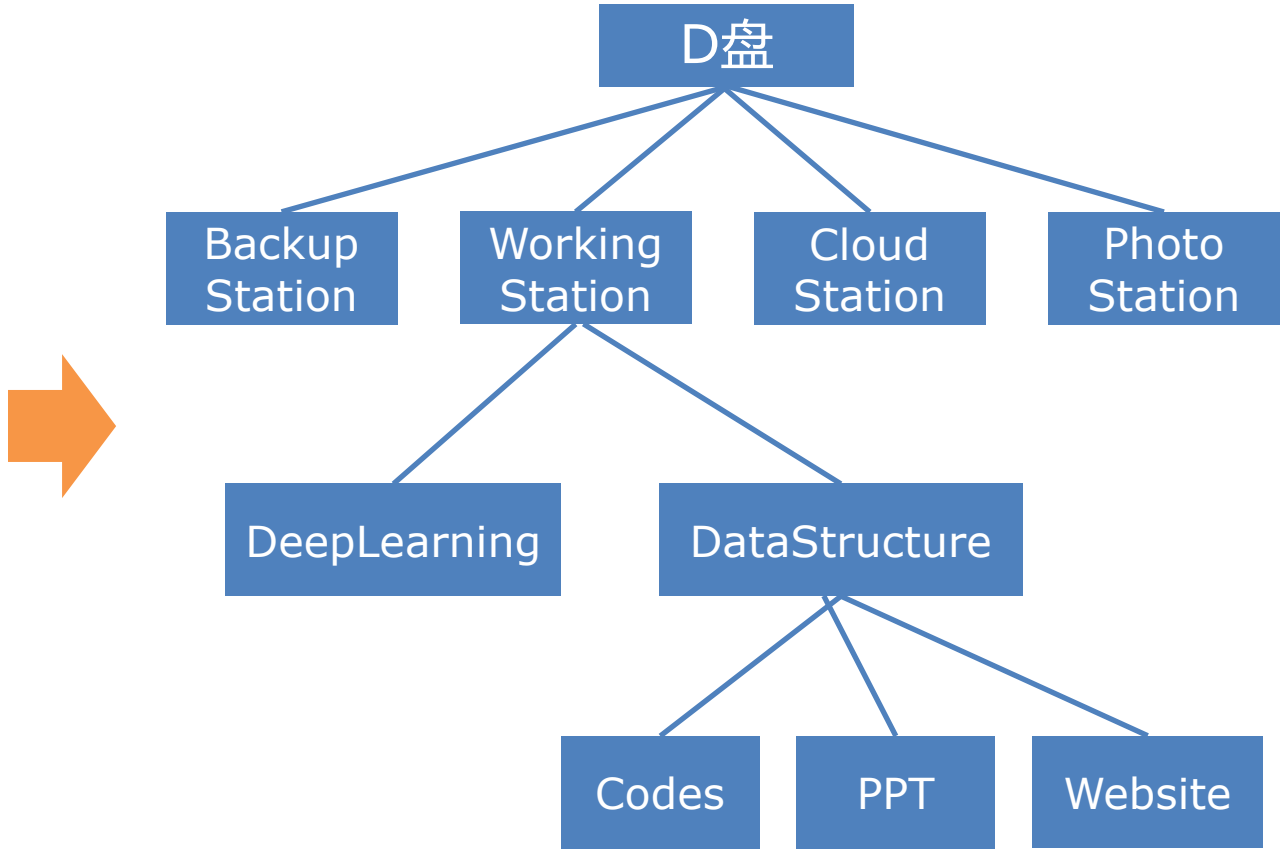
什么是树


十八世
十九世
廿世
廿一世
廿二世
廿三世
廿四世
汝举 氏
二公 思□ 潘氏
大公 思元 韦氏
二庆三 氏
大玉三 胡氏
三怀荣 张氏
二星三 施氏
大得三 吴氏
二锡财 吴翠英
大锡宽 薛云生
金成 吕氏
二夕忠 钱宏香
大夕有 夏氏
二殷有 时文香
大殷臣 王氏
宪 姜梅
乔英 乔林 张铁梅
殷英
二殷林 崔氏
大殷银 冯氏
二殷发 赵氏
大殷才 朱氏
旭亚 氏
兵 陶慧萍
罗罗
二文明 殷玉霞
大志兵 王怀琴
小燕 家成 氏
春梅 家骆 氏
青云 青香
齐瑞 氏
福建 氏
立平 莉莉
住乔田乔田
住蚌埠市
住永丰酒店婆庄
住大通果树
住大通坝田刘镇

树的基本概念

什么是树


DataStructure
DataStructure
在 DataStructure 中搜
新建
排序
预览
Data (D:)
BackupStation
CloudStation
PnermanentlyStoring
WorkingStation
_OLDCourse
ComputerMath
ComputerVision
ComputerVision1X
Datasets.ouxinyu.cn
DataStructure
.vscode
Codes
PPT
PPT（雨课堂）
Website
参考资料
DeepLearning
名称
修改日期
类型
.vscode 2023/9/21 23:01 文件夹
Codes 2023/9/25 15:34 文件夹
PPT 2023/10/4 12:03 文件夹
PPT（雨课堂） 2023/9/13 19:40 文件夹
Website 2023/9/24 21:01 文件夹
参考资料 2023/8/25 20:13 文件夹
教学进度计划 2023/8/20 19:47 XLSX
数据结构_教学进度计划（64课时） 2023/7/22 0:10 DOCX

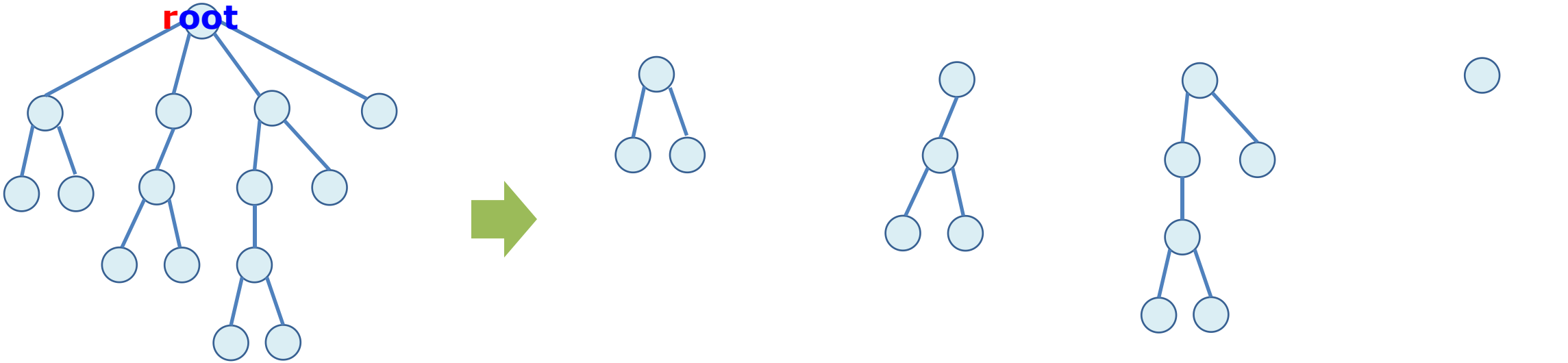

D盘
Backup Station
Working Station
Cloud Station
Photo Station
DeepLearning
DataStructure
Codes
PPT
Website

树的基本概念

树的定义

树（Tree）是 n(n≥0) 个结点构成的有限集合。当 n=0 时，称为空树；对于任意一颗非空树 (n > 0)，它具备如下性质：

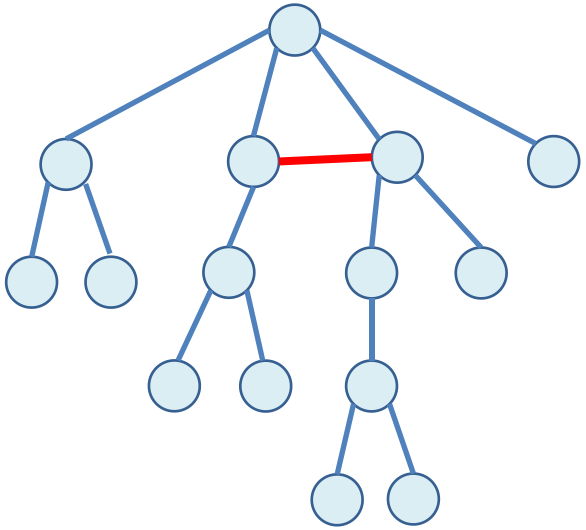
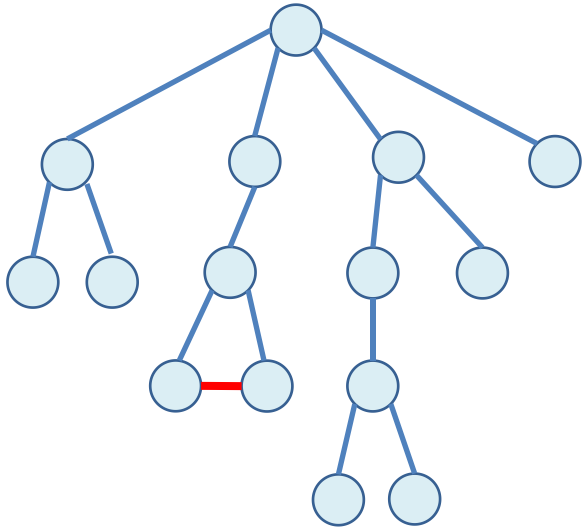
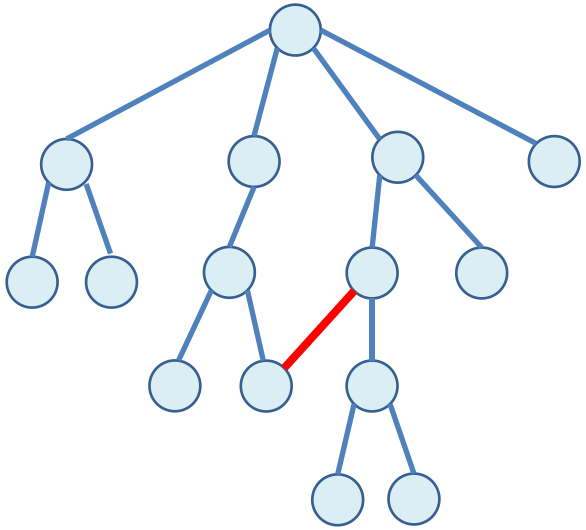
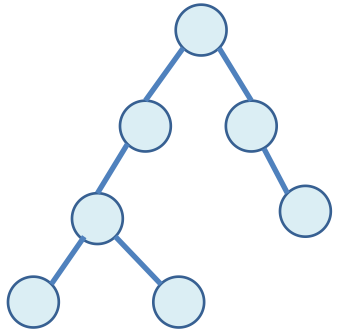
- 有且仅有一个称为根（root）的特殊结点，用 r 表示
- 除根结点外的其余节点可分为 m(m > 0) 个互不相交的有限集T_1, T_2,..., T_m，其中每个集合本身又是一棵树，称为原树的子树（SubTree）



(A) 树 T　(B) 树 T_1　(C) 树 T_2　(D) 树 T_3　(E) 树 T_4

树的基本概念

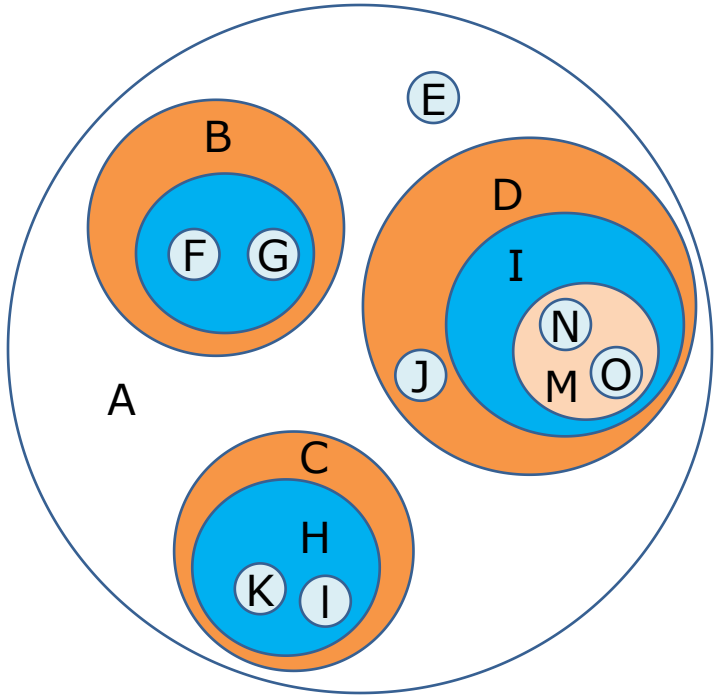
树与非树?

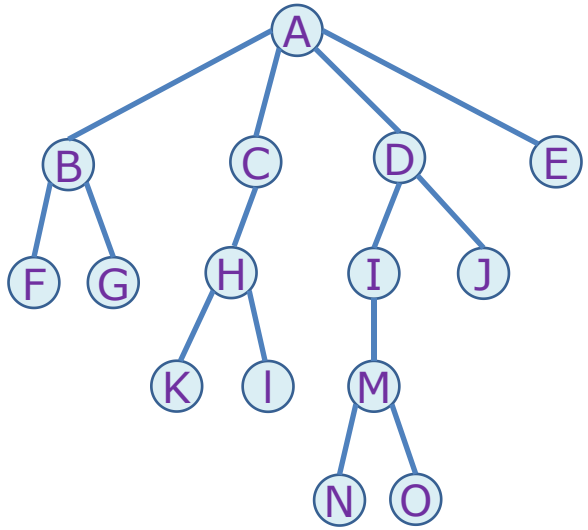
- ✓ 子树是不相交的
- ✓ 除了根结点外，每个结点有且仅有一个父结点
- ✓ 一颗包含N个结点的树，有N-1条边

树的基本概念

树的表示方法

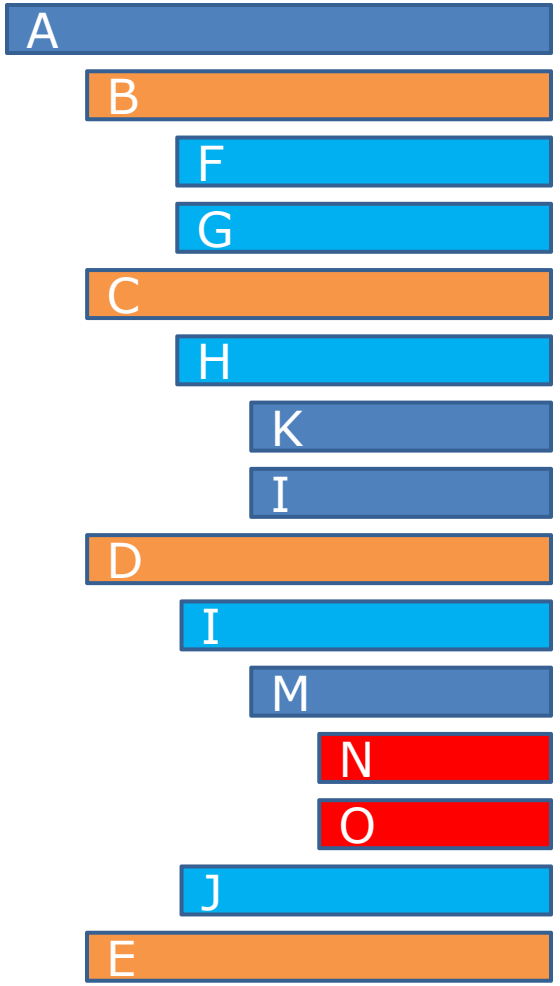


嵌套集合



(A(B(F,G), C(H(K, I)), D(I(M(N, O)), J), E))

广义表

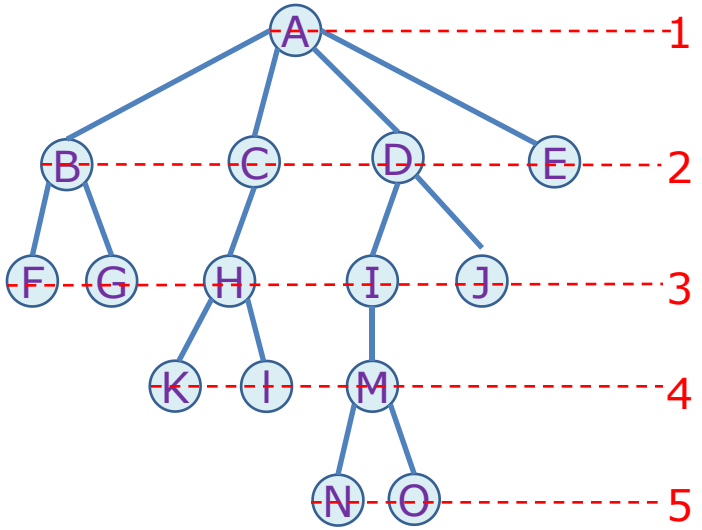


凹入表示

树的基本概念

基本术语

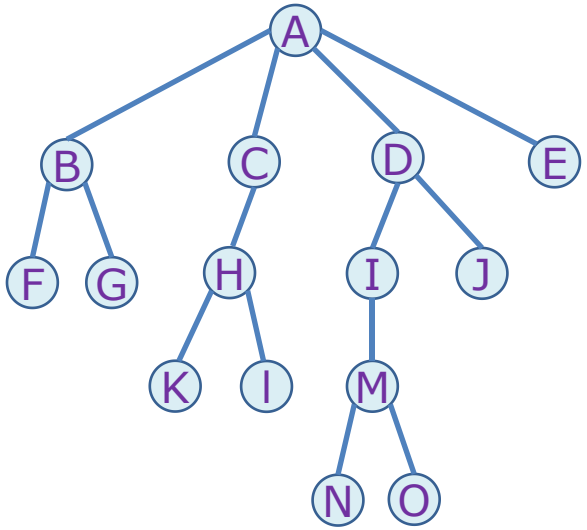
- **结点（Node）**：树中的元素，结点是树的基本构成部分（A、B、C）
- **边（Edge）**：连接两个结点的有向线段，表示它们之间的关系，指向结点的边称为入边，指向外部的边称为出边。（AB、AC、BG...）
- **结点的度（Degree）**：结点非空子树个数（A的度为4，B的度为2）
- **树的度：**所有结点度的最大值称为树的度（右边树的度为4）；树中所有结点的度的和等于结点数-1，即$n_d=n-1$。
- **结点的层次（Level）**：规定根结点在第1层，其他任意结点的层数是其父结点的层数加1（根节点A的层数为1，结点N的层数为5）
- **树的深度/高度（Depth/Height）**：树中所有结点中的最大层次是这棵树的深度（右边树的深度为5）



树的基本概念

基本术语

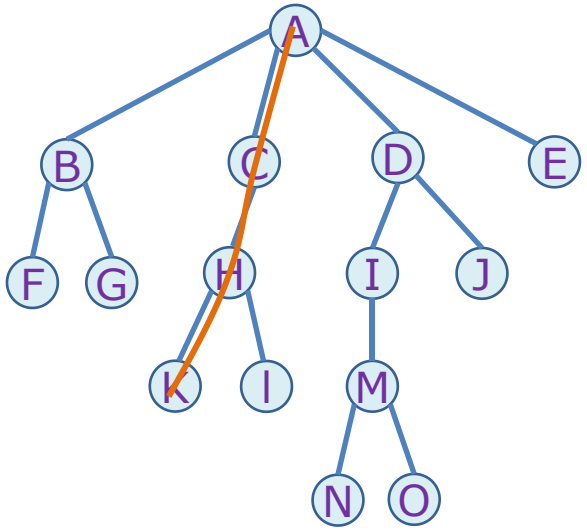
- **根结点（Root）**：树中唯一没有入边（前驱）的结点（A）
- **叶子结点（Leaf）**：度数为0的结点（F、G、K、I、N、E...）
- **森林（Forest）**：m 个不相交的树的集合（删除A后的四棵子树的集合）
- **父结点（Parent）**：又称为双亲。若某结点有子树，则该结点称为子树的父结点。（B是F、G的父结点，D是I、J的父结点）
- **子结点（Child）**：若A是B的父结点，则称B是A的子节点



树的基本概念

基本术语

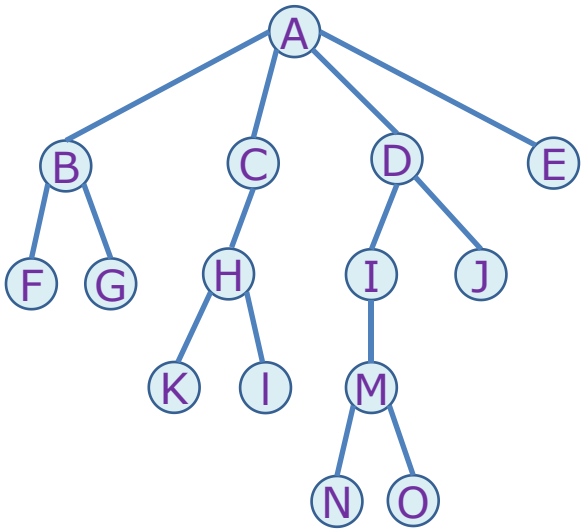
- **兄弟结点（Sibling）**：具有同一父结点的个结点彼此是兄弟结点
  （BCDE是兄弟、FG是兄弟、H没有兄弟、IJ是兄弟）
- **堂兄弟结点（Cousin）**：父结点位于同一层，但不是同一个父结点的结点（F和H，G和I，H和J）
- **祖先结点（Ancestor）**：沿树根到某一结点路径上的所有结点都是该节点的祖先结点（ACH都是K的祖先，ADIM都是O的祖先）
- **子孙节点（Descendant）**：某一结点的子树中的所有结点是该结点的子孙（树中所有的结点都是根节点A的子孙，HKI是C的子孙）
- **路径（Path）**：从某结点沿树中的边可到达另一个结点，则两个结点之间的边的集合称为路径，路径所包含边的个数称为路径的长度。
  （AK的路径为AC->CH->HK，其长度为3，AN的长度为4）



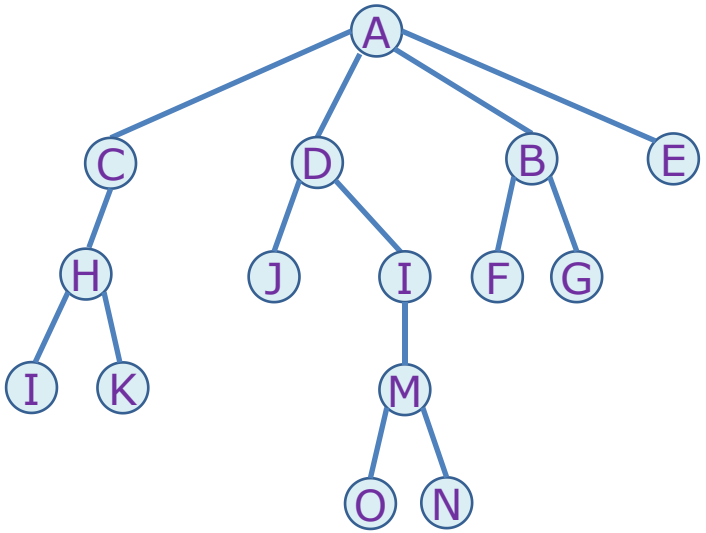
树的基本概念

基本术语

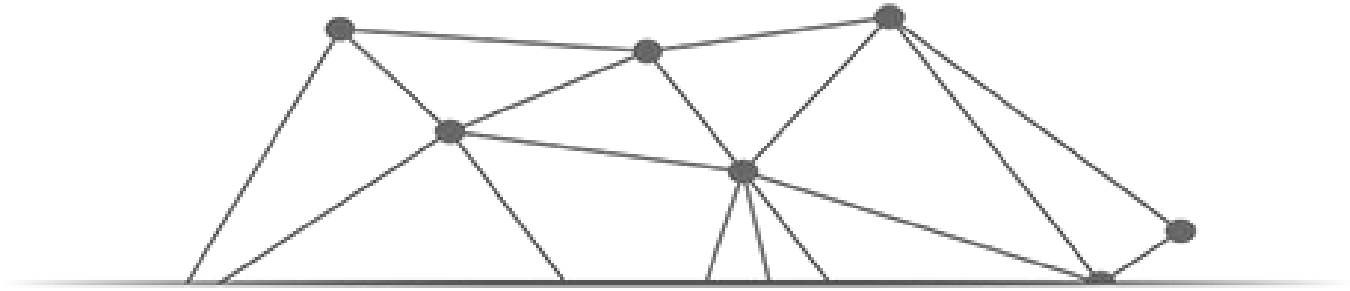
- **无序树**：树中结点的各子树之间的顺序无关，可以任意交换位置。若T1=T2，则T1是无序树
- **有序树**：树中结点的各棵子树从左到右顺序相关，交换任意两棵子树的位置都将改变原来树的结构。
  若T1≠T2，则T1是有序树



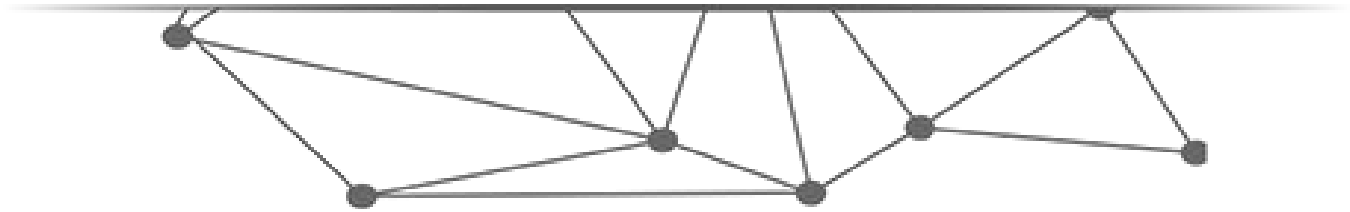
树 T_1

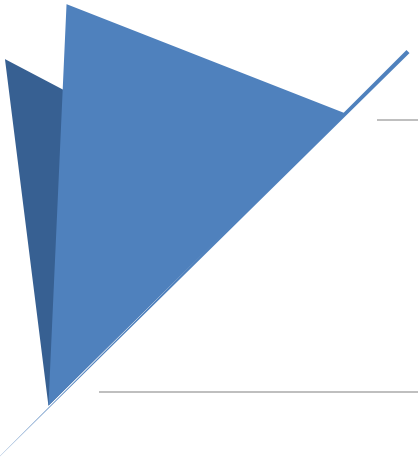


树 T_2



课堂互动 10.1





二叉树的基本概念

/ 二叉树的定义

/ 斜二叉树

/ 满二叉树

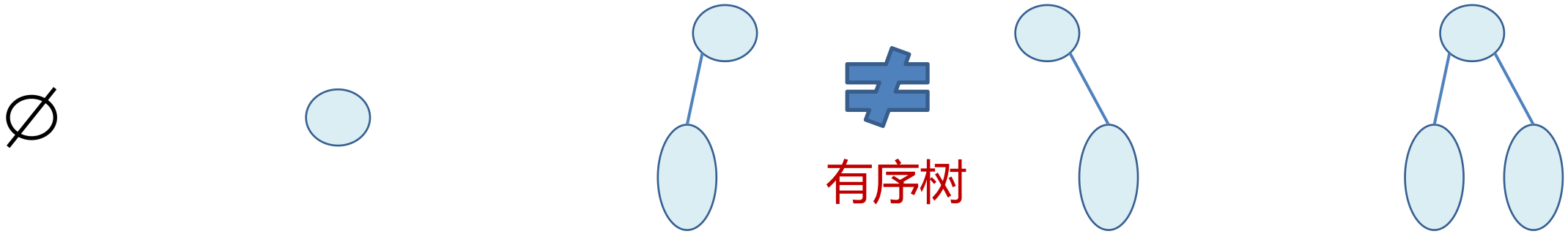
/ 完全二叉树

二叉树的基本概念

二叉树的定义

二叉树（Binary Tree）是结点的有限集合。该集合可以是空，也可以是由根结点和称为其左子树和右子树的两个不相交结点所构成的两个分支的树。

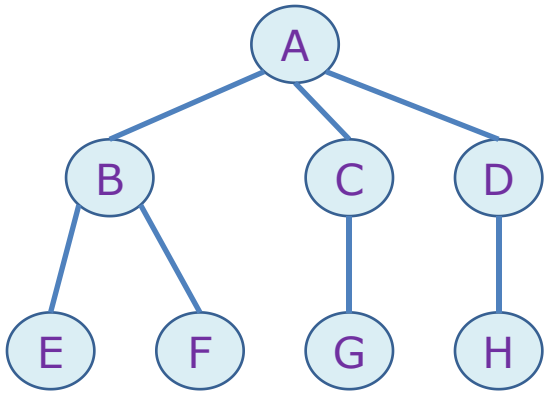
- **二叉树的五种基本形态**

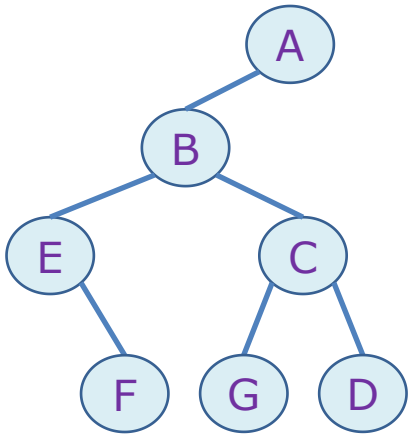


(a) 空树　(b) 只有根结点　(c) 只有左子树，但右子树为空　(d) 只有右子树，但左子树为空　(e) 包含左右两个子树

二叉树的基本概念

树与二叉树的区别



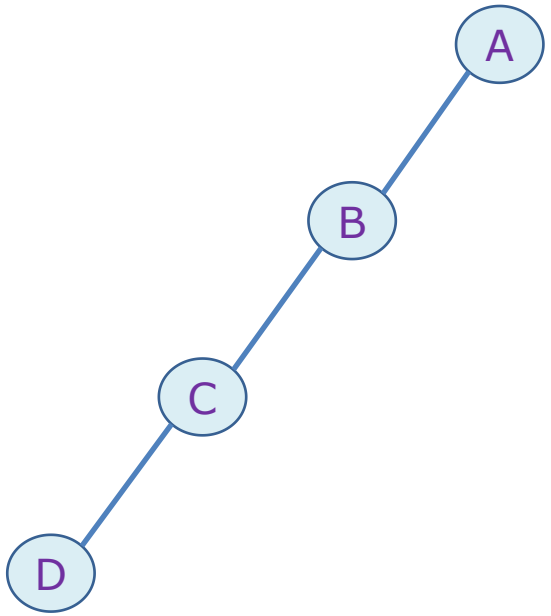


- 树中结点的子树之间可以是无序的，而二叉树中结点的子树即使只有一棵子树，也要区分左、右子树。
- 树中每个结点可以有 0 棵或多颗子树，而二叉树的每个结点最多只能有2棵子树。

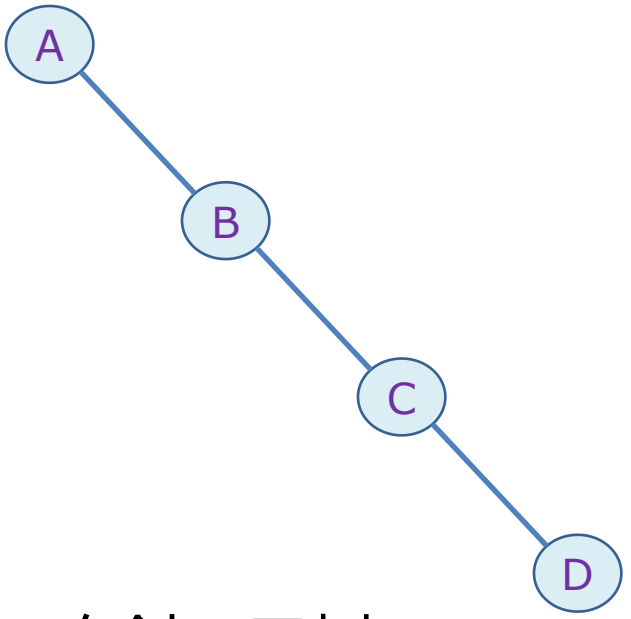
特殊二叉树

斜二叉树

只向一边(左边或右边)生长的二叉树，称为**斜二叉树（Skewed Binary Tree）**。



左斜二叉树



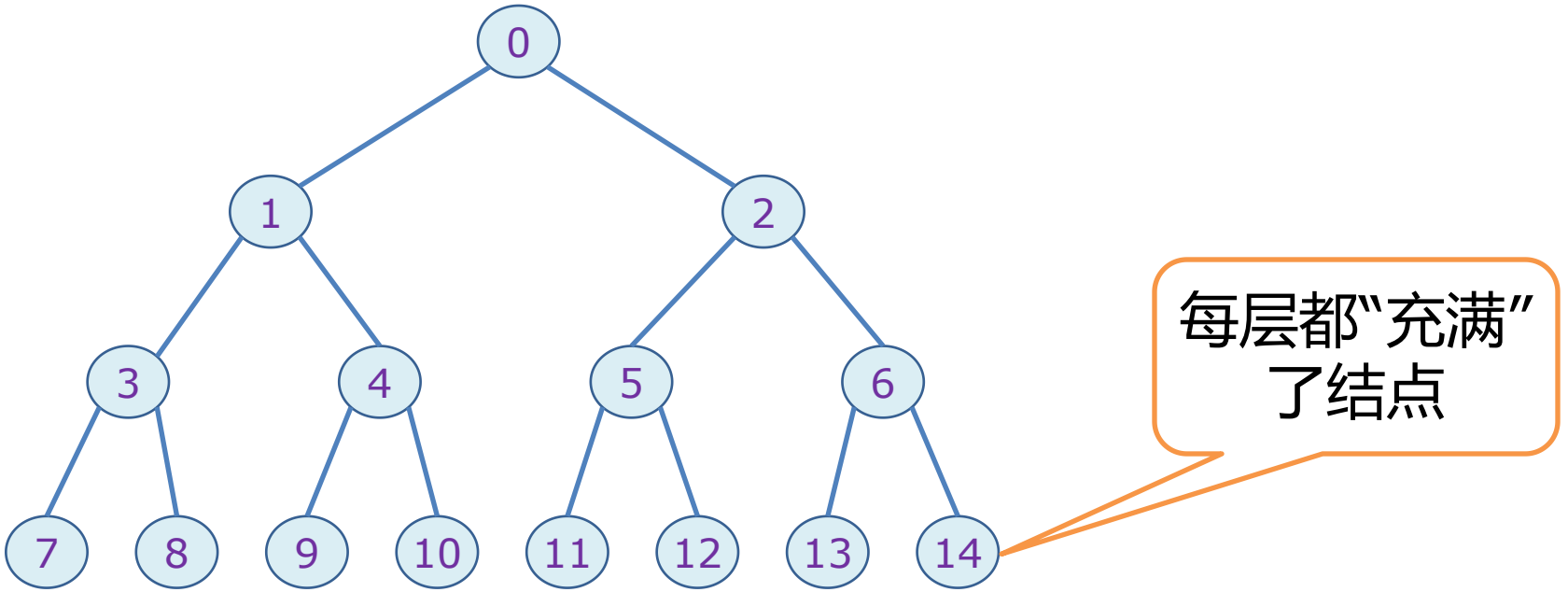
右斜二叉树

斜二叉树是深度最高的二叉树之一

特殊二叉树

满二叉树

满二叉树（Full Binary Tree），又称为**完美二叉树（Perfect Binary Tree）**，若其高度为h，则该二叉树包含2^h-1个结点。即每一层的结点数量都达到饱和状态的二叉树。



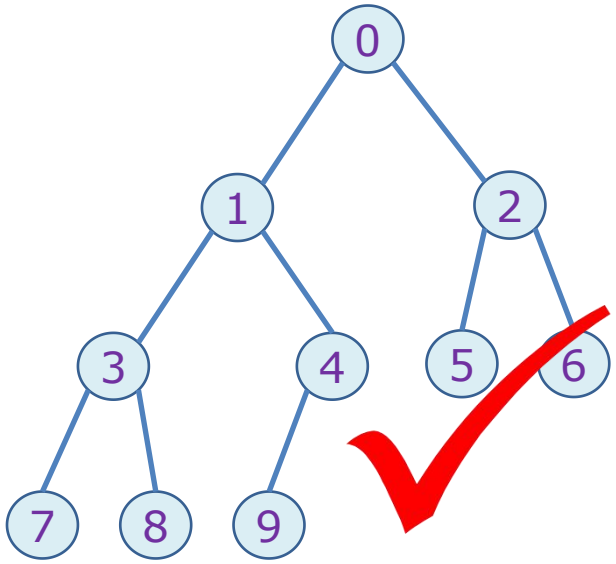
满二叉树是深度最低的二叉树

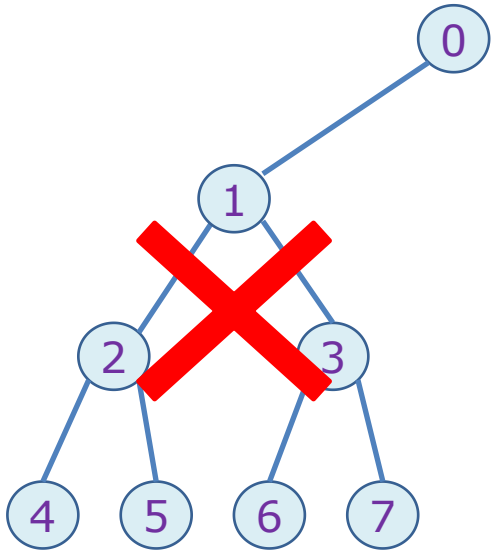
特殊二叉树

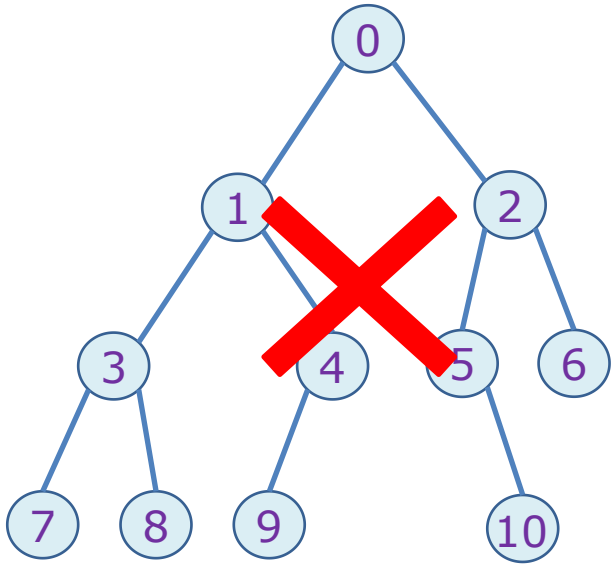
完全二叉树

深度为k，有n个结点的二叉树，当且仅当其每个结点都与深度为k的满二叉树中编号从1至n的结点一一对应时，称为**完全二叉树（Complete Binary Tree）。完全二叉树具有以下两个性质：**

- 只有最下面两层结点的度可以小于2，其他层结点度都等于2
- 最下面一层的叶子结点均依次集中在靠左的若干个位置上
- 度数为1的结点最多只有1个，最少为0个



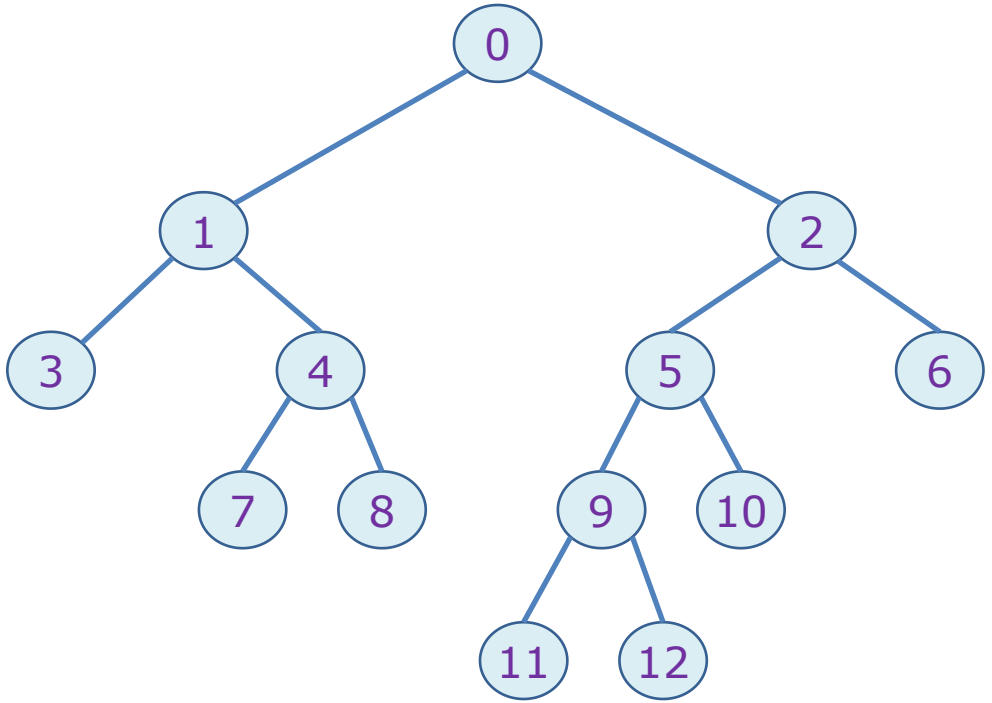




特殊二叉树

扩充二叉树

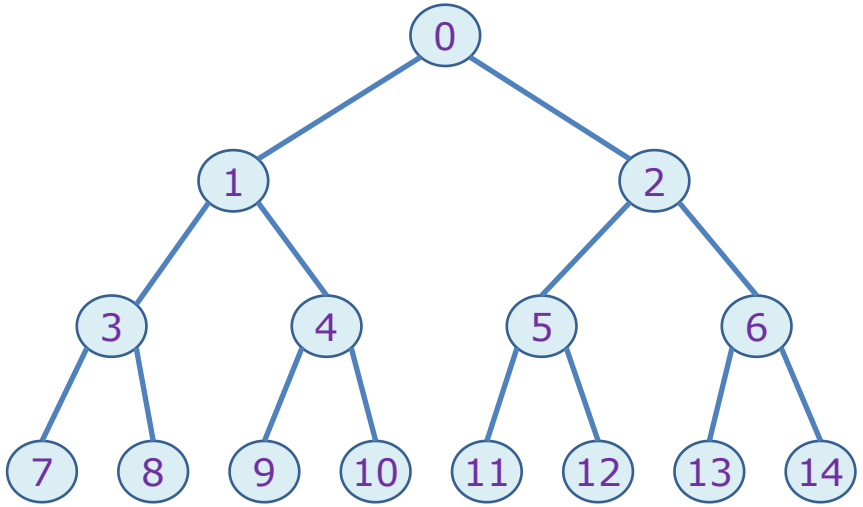
除叶子结点外，其余结点都必须有2个孩子结点的二叉树称为**扩充二叉树(Extended Binary Tree)**，也称为**2-树**。



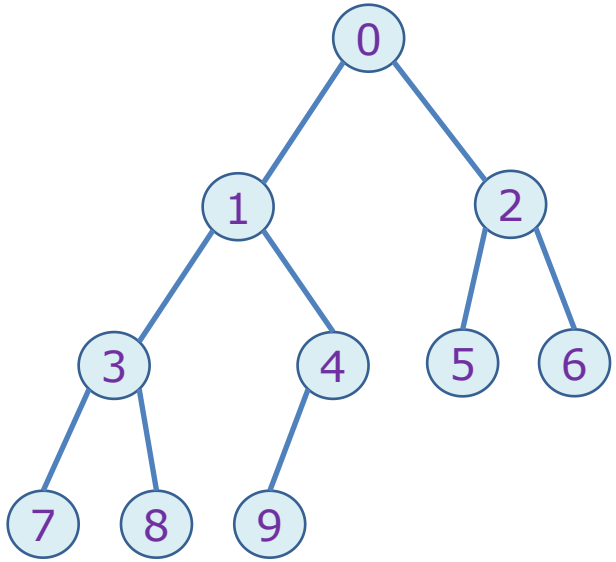
- ✓ 仅存在度为2和0的结点
- ✓ 不存在度为1的结点

特殊二叉树

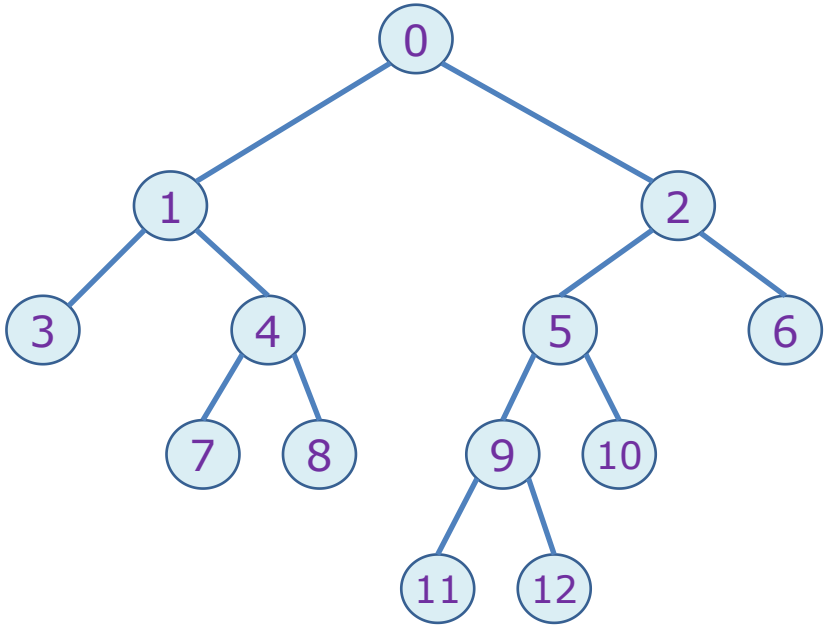
扩充二叉树



满二叉树 + 完全二叉树 + 扩充二叉树

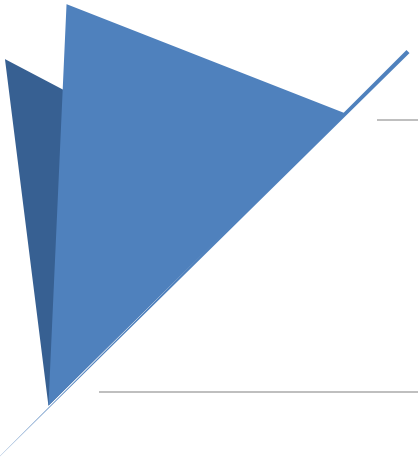


完全二叉树



扩充二叉树

满二叉树一定是完全二叉树，也一定是扩充二叉树



二叉树的性质

/ 第i层结点数量

/ 所有结点的数量

/ 高度

/ 度数

二叉树的性质

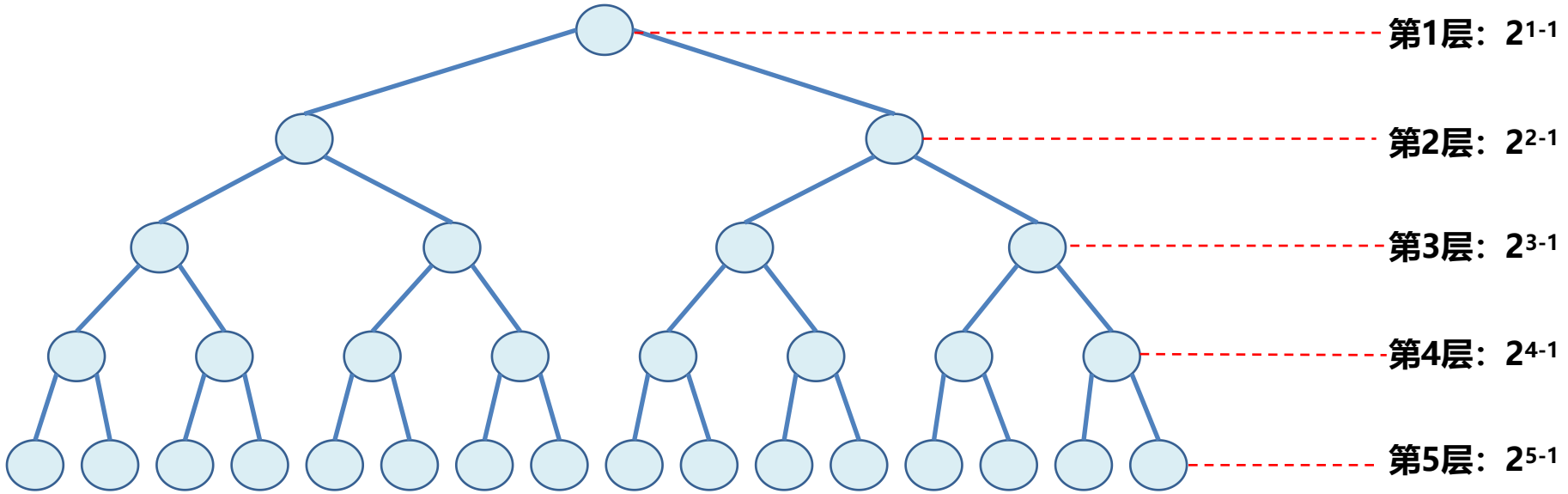
【性质1】二叉树的第i(i≥1)层上至多有 2^{i-1} 个结点。

提问：第i层上至少有 1 个结点?

证明：（数学归纳法）

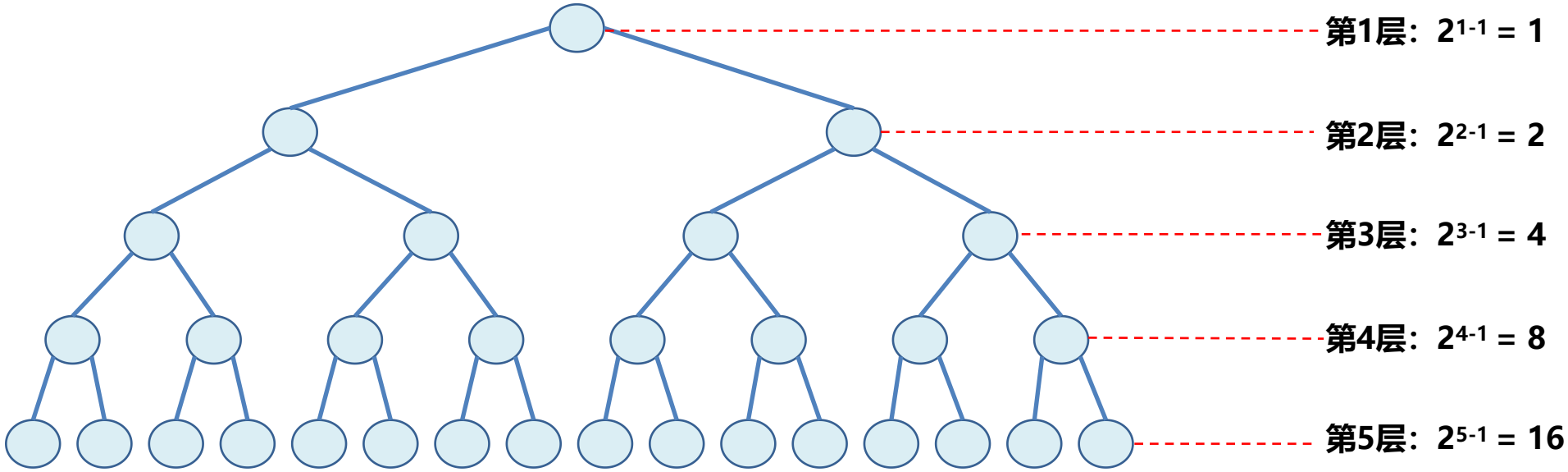
1. 当 i = 1 时，二叉树至多只有一个结点，性质1成立。
2. 假设 i = k 时结论成立，则有二叉树第 k 层上至多有 2^{k-1}个结点；
3. 当 i = k+1时，因为每个结点最多只有两个孩子，所以第 k+1 层上至多$2*2^{k-1}$个结点，性质1成立。

   综上可知，性质1成立。



二叉树的性质

【性质2】高度为h的二叉树上至多有 2^h-1 个结点，此时为满二叉树。



证明：（数学归纳法）

1. 当 h = 0 时，二叉树为空二叉树，性质2成立。
2. 当 h > 0 时，利用性质1，高度为h的二叉树中结点总数最多为：

$$\sum_{i=1}^{h} 2^{i-1} = (2^0 + 2^1 + 2^2 + \ldots + 2^{h-1}) = \frac{1(1-2^h)}{1-2} = 2^h - 1$$

提问：深度为k时至少有 **k** 个结点?

提问：深度为k的完全二叉树至少有 2^{k-1} 个结点?

单选题 1分

设置

例1：若一棵二叉树有126个结点，则在第7层至多有多少个结点?

A 32

B 64

C 63

D 不存在第7层

提交

二叉树的性质

【性质3】包含n个结点的二叉树的高度为 $\lceil \log_2 (n+1) \rceil$ 或 $\lfloor \log_2 n \rfloor + 1$。

$$2^{k-1} - 1 < n \leq 2^k - 1$$

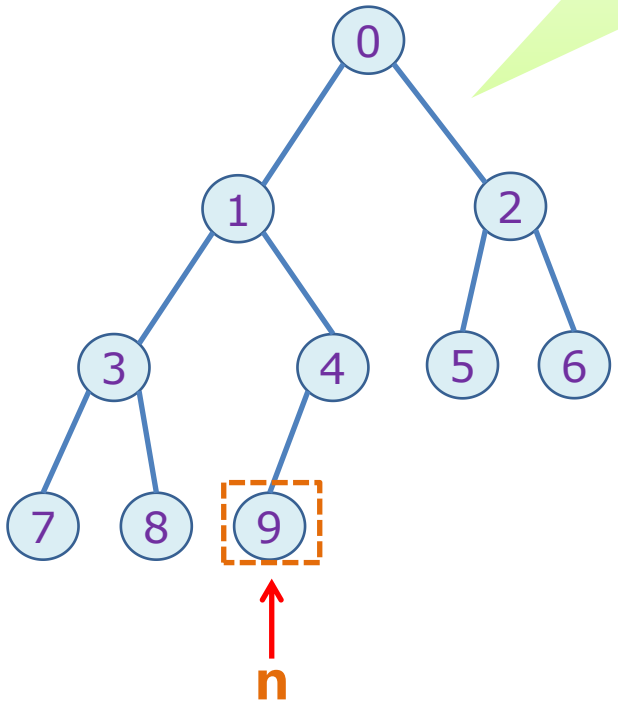
$$2^{k-1} \leq n < 2^k$$

$$k - 1 < \log_2 (n+1) \leq k$$

$$k - 1 \leq \log_2 n < k$$

$$\lceil \log_2 (n+1) \rceil \leq k \leq \lfloor \log_2 n \rfloor + 1$$

提问：高度最高为__n__（_斜二叉树_）？



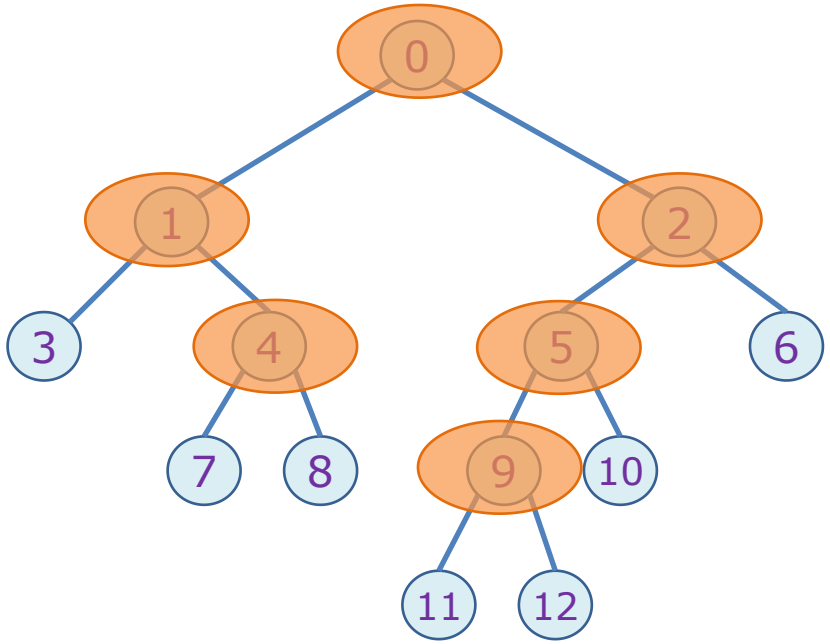
第k-1层：$2^{k-1}-1$

第k层：2^k-1

二叉树的性质

【性质4】任意非空二叉树，若叶子数（度数=0）为n_0，度数为2的结点数为n_2，则有：

$$n_0 = n_2+1$$



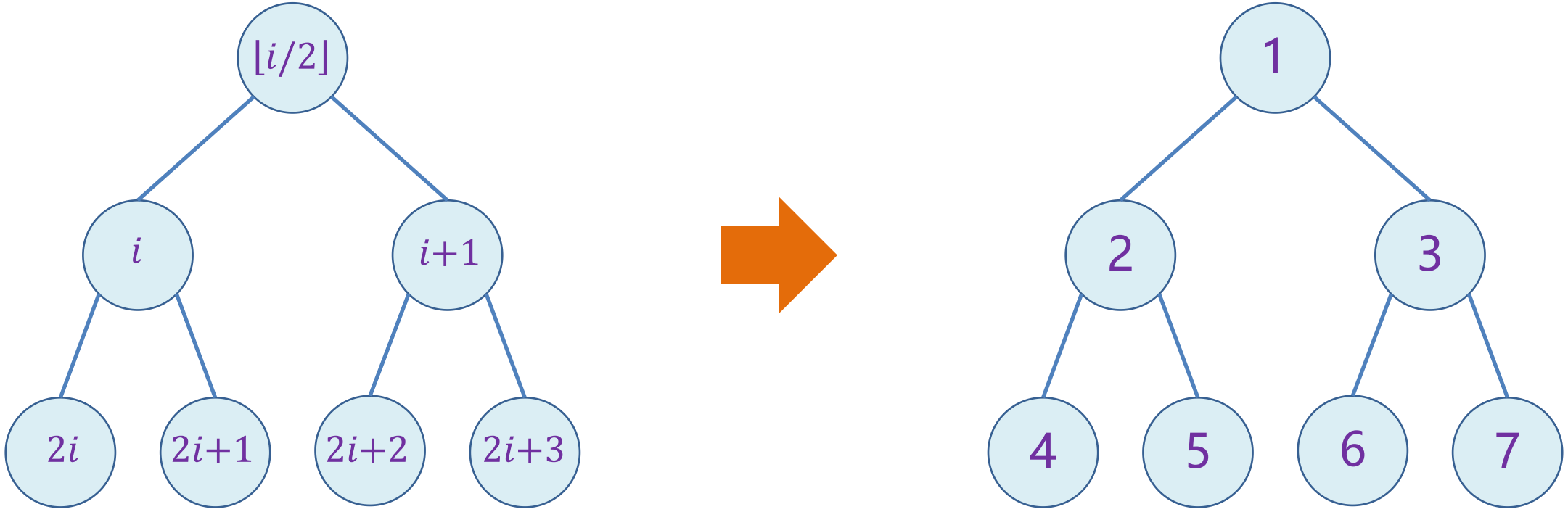
$n_2 = 6$

$n_0 = 7$

$n_0 = n_2 + 1$

二叉树的性质

【性质5】对于完全二叉树，若从上至下、从左至右编号，则编号为 i 的结点，其左孩子编号必为 2i，其右孩子编号必为 2i+1，其双亲结点的编号必为 $\lfloor i/2 \rfloor$ 。



单选题 1分

设置

例2：一棵完全二叉树有8000个结点，试求该二叉树叶子结点的个数是多少?

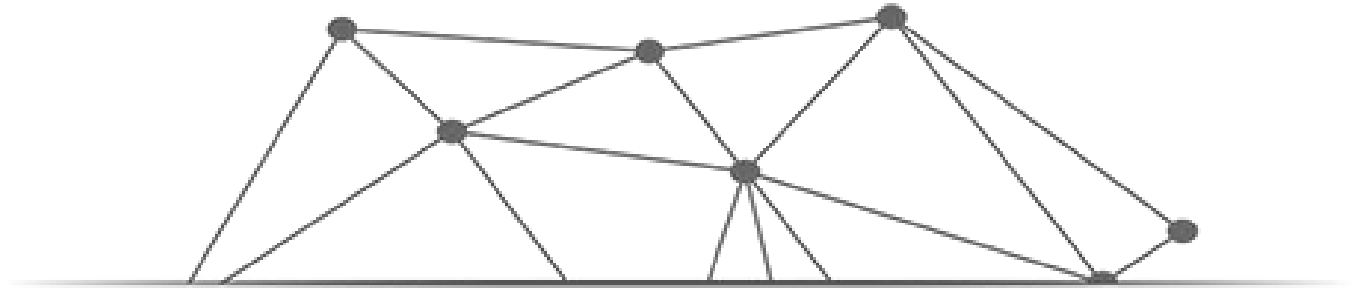
A 3999

B 4000

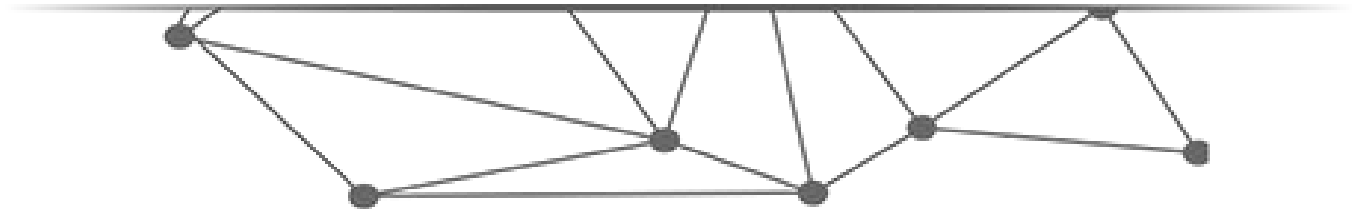
C 4001

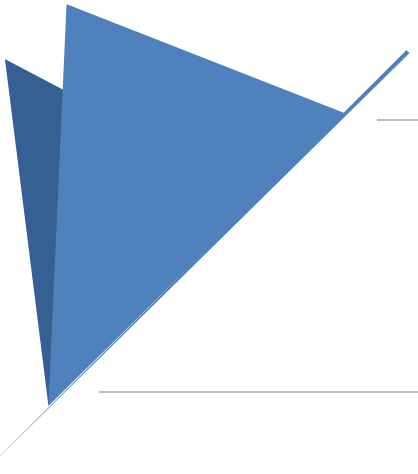
D 无法计算

提交



课堂互动 10.2





二叉树的存储结构

/ 二叉树的抽象数据类型

/ 二叉树的顺序存储

/ 二叉树的链式存储（孩子表示法）

/ 二叉树的三叉链表（孩子+双亲）

二叉树的存储结构

二叉树的抽象数据类型定义

ADT BinaryTree {

数据对象D：是具有相同特性的数据元素的集合

数据关系R:

1. 若D=∅，则R=∅，则称为空二叉树。
2. 若D≠∅，则R={H}，H具有如下二元关系。
   1). 在D中存在唯一的称为根的数据元素root，它在关系H中无前驱；　// 关于根的说明
   2). 若D-{root}=∅，则存在D-{root}={D_1, D_r}，且$D_1 \cap D_r = \emptyset$；　// 关于树不相交的说明
   3). ...　// 关于数据元素的说明
   4). ...　// 关于左、右子树的说7明

ADT BinaryTree {

数据对象D：是具有相同特性的数据元素的集合

数据关系R:

基本操作P:

InitBiTree(&T)：构造空二叉树T

CreateBiTree(&T, definition)：按definition构造二叉树

Assign(T, &e, value)：将二叉树T的结点e赋值为value

LeftChild(T, e)：返回二叉树T中结点e的左孩子

LeftSibling(T, e)：返回二叉树T中结点e的左兄弟

...

PreOrderTraverse(T)：先序遍历二叉树T的每一个结点

InOrderTraverse(T)：中先序遍历二叉树T的每一个结点

PostOrderTraverse(T)：后先序遍历二叉树T的每一个结点

LevelOrderTraverse(T)：层序遍历二叉树T的每一个结点

} **ADT BinaryTree**

二叉树的存储结构

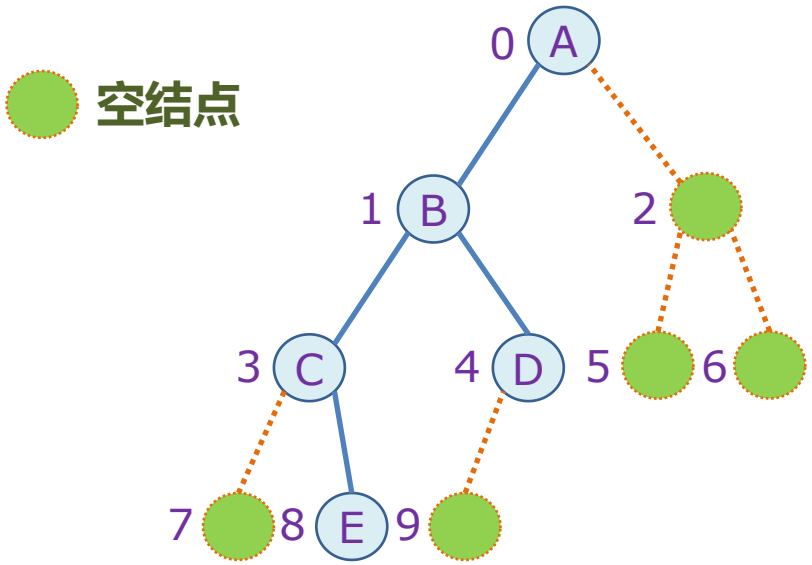
二叉树的顺序存储

结点间关系蕴含在存储位置中

- **存储规则：**

1. 在二叉树中增加“空结点”，将一般二叉树补全为完全二叉树。
2. 将包含“空结点”的“完全二叉树”用数组进行顺序存储。

- **结点结构：**



序号	0	1	2	3	4	5	6	7	8	9
结点	A	B	^	C	D	^	^	^	E	^

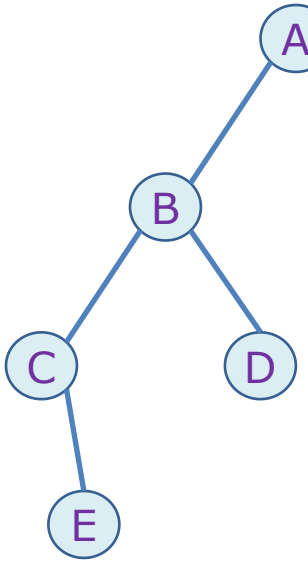
理论上可行，但可能存在较大的空间浪费
实际应用中，顺序存储在二叉树中并不适用

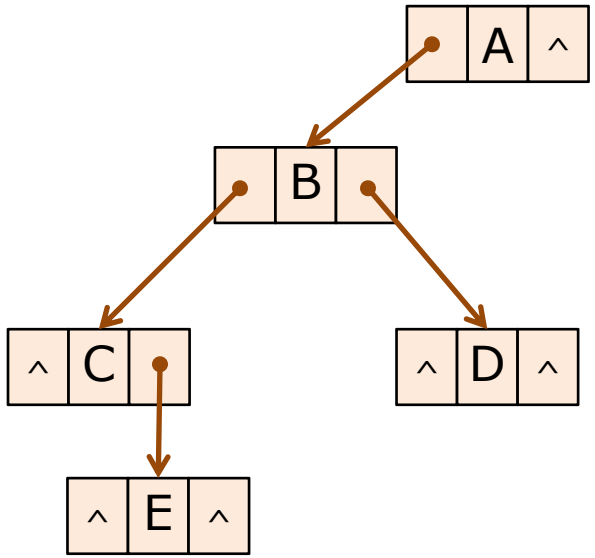
二叉树的存储结构

二叉树的链式存储（孩子表示法）

- **存储规则：**每个结点中包含一个数据域、两个指针域（左孩子、右孩子）
- **结点结构：**

lChild	data	rChild





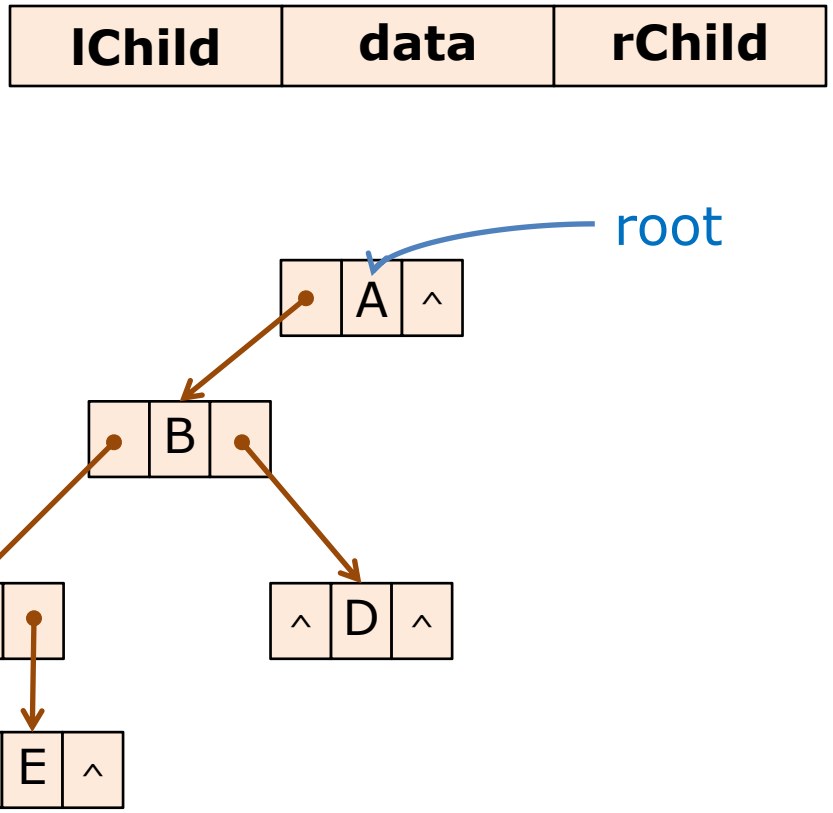
- **优点：**有利于从父结点到孩子方向进行访问
- **缺点：**从孩子结点到父结点访问较困难

二叉树的存储结构

二叉树的链式存储

二叉树的链式存储结构定义

```
typedef struct BTNode {
    ElemType data;
    struct BTNode *lChild;
    struct BTNode *rChild;
} BTNode

typedef struct BinaryTree {
    BTNode *root;
} BinaryTree
```

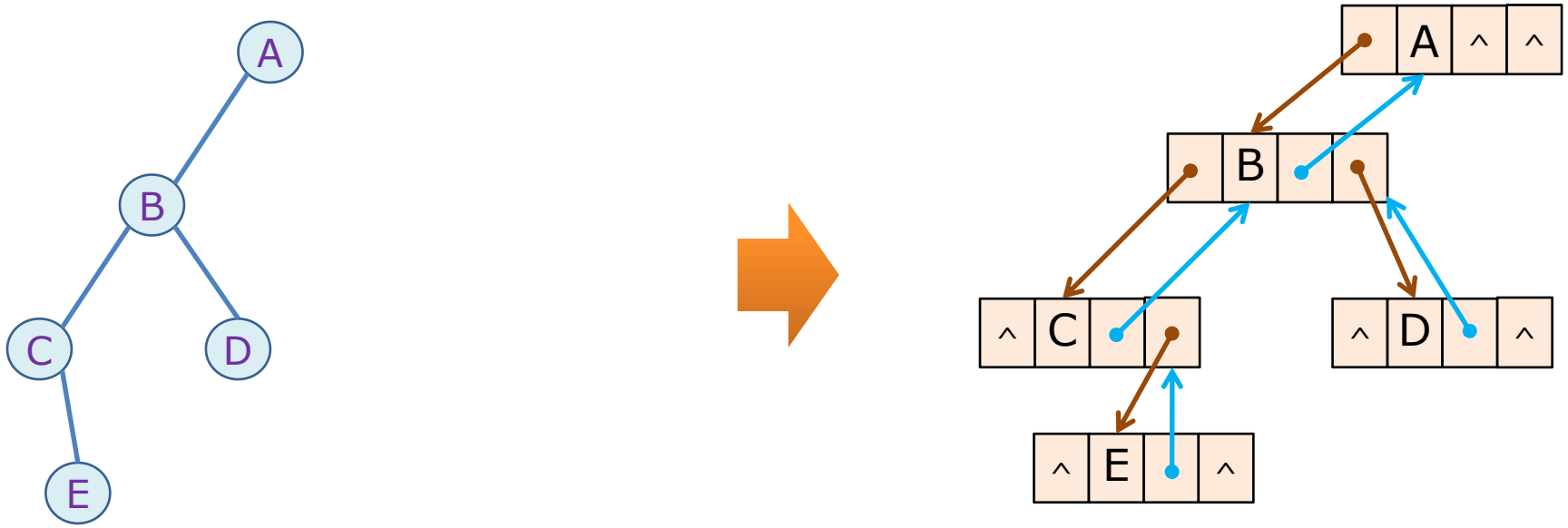


二叉树的存储结构

二叉树的链式存储（三叉链表）

- **存储规则：**在二叉链表的基础上增加一个parent域，指向该结点的父结点。
- **结点结构：**

lChild	data	parent	rChild



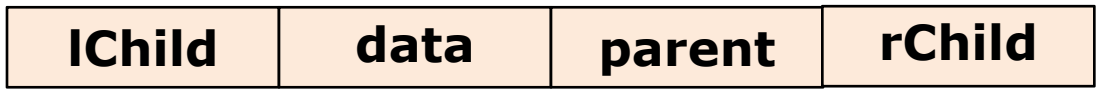
- **优点：**既可以从父结点到孩子结点进行访问，又可以从孩子结点到父结点进行访问

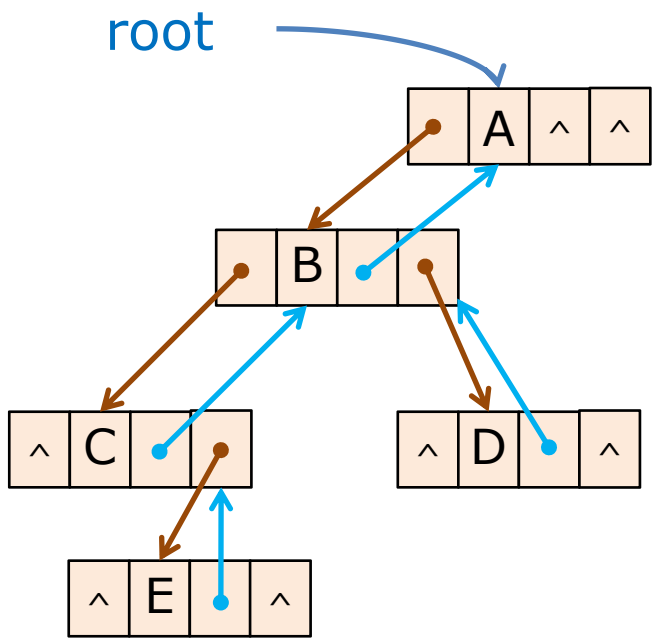
二叉树的存储结构

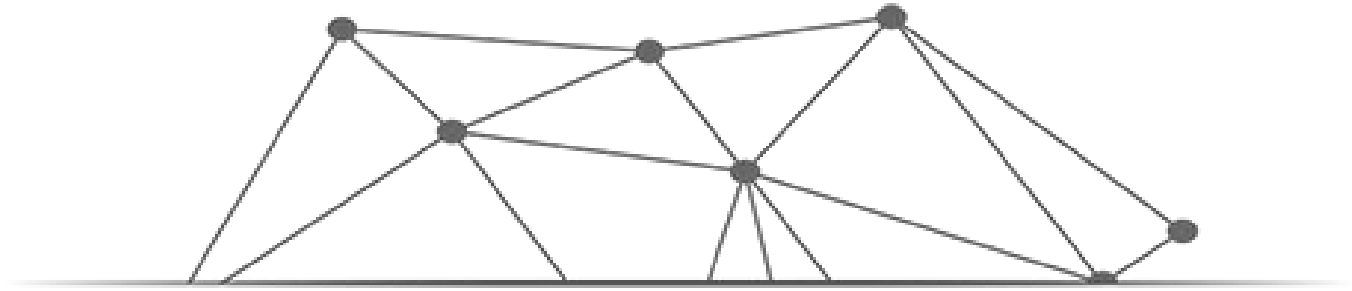
二叉树的链式存储（三叉链表）

三叉链表存储结构定义

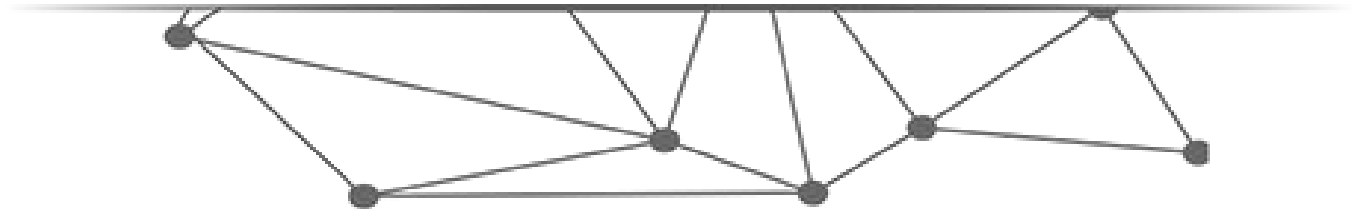
```
typedef struct BTNode {
    ElemType data;
    struct BTNode *lChild;
    struct BTNode *rChild;
    struct BTNode *parent;
} BTNode

typedef struct BinaryTree {
    BTNode *root;
} BinaryTree
```







课堂互动 10.3



第10讲 树和二叉树的基本概念

小　节

- 二叉树是一种常用的树结构，它具有一些特别而且有用的性质
- 满二叉树、完全二叉树、扩充二叉树是最常用的两种特殊形态二叉树
- 二叉树有顺序和链式两种存储表示。
- 顺序存储将所有节点按照层次顺序存储到连续的存储单元，这种方法更适合完全二叉树；链式存储又称二叉链表，每个结点包含2个指针，分别指向左孩子和右孩子，这种方法适用于一般二叉树。
- 为解决二叉链表对父结点访问的限制，增加一个parent用于保存父节点的地址，这种方法称为三叉链。

欧老师的联系方式

读万卷书 行万里路 只为最好的修炼

QQ：14777591 (宇宙骑士)

Email：ouxinyu@alumni.hust.edu.cn

Website: http://ouxinyu.cn

Tel：18687840023

地址：安宁校区 诚远楼201

南院 智能应用研究院A306-2