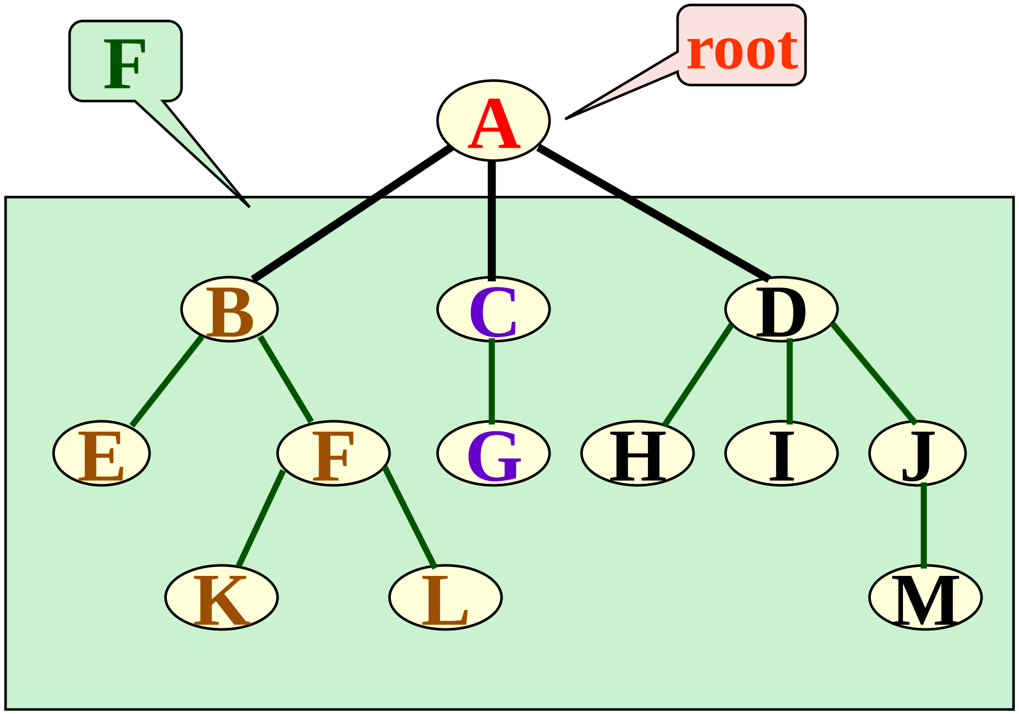
第6章 树

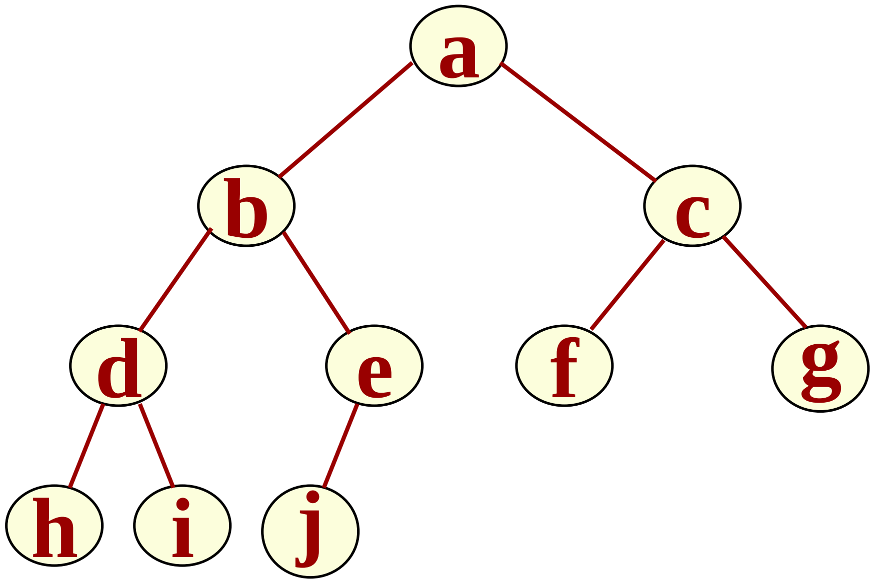
对比树型结构和线性结构的结构特点

线性结构	树型结构
第一个数据元素 (无前驱)	根结点 (无前驱)
最后一个数据元素 (无后继)	多个叶子结点 (无后继)
其它数据元素 (一个前驱、一个后继)	其它数据元素 (一个前驱、多个后继)

森林(forest):

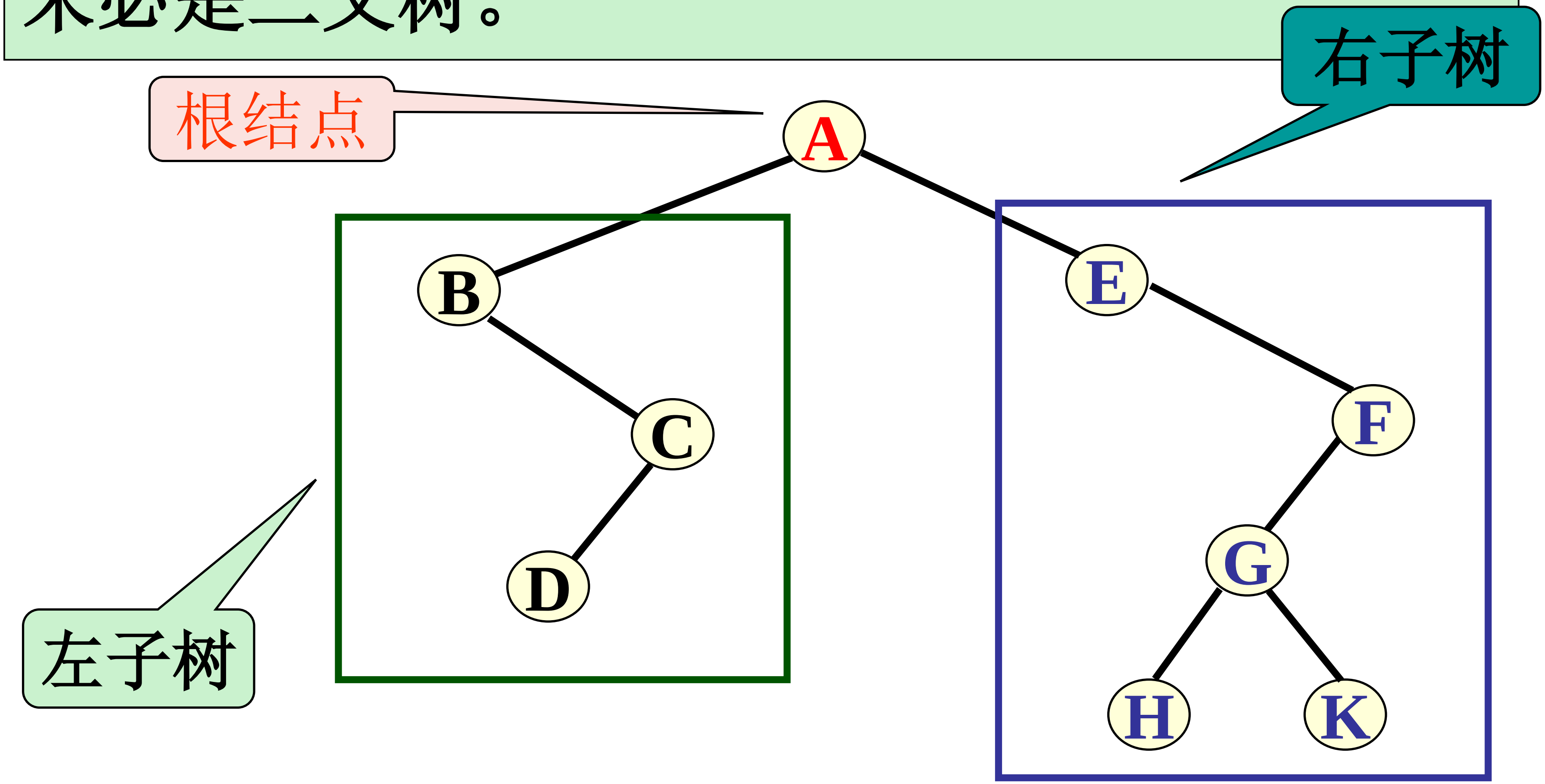
是m(m≥0)棵互不相交的树的集合



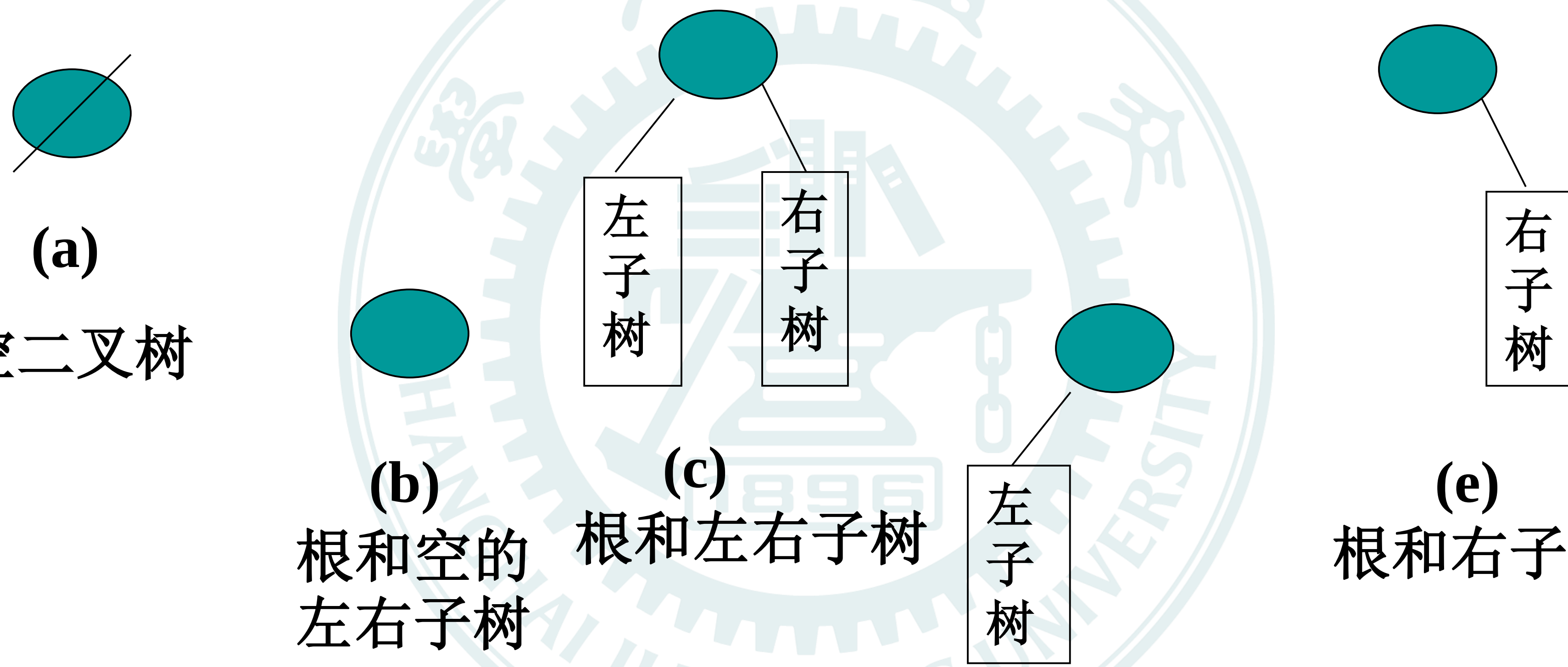


树的结点数n和分支数b的关系是： b=n-1

注意: 二叉树是**有序树**,它的子树有左右之分。**二叉树的度数不超过二,但度数不超过二的树未必是二叉树。**



二叉树的基本形态



(a) 空二叉树

(b) 根和空的左右子树

(c) 根和左右子树

(d) 根和左子树

(e) 根和右子树

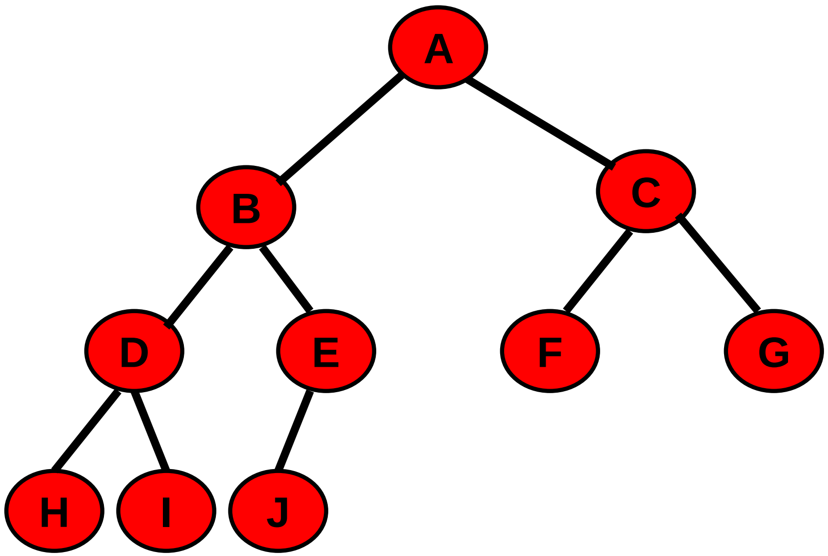
完全二叉树

在满二叉树的最底层自右至左依次(注意：不能跳过任何一个结点)去掉若干个结点得到的二叉树也被称之为完全二叉树。满二叉树一定是完全二叉树，但完全二叉树不一定是满二叉树。

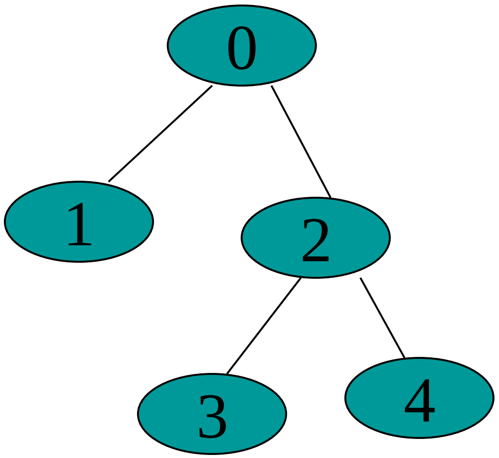
完全二叉树的特点是：

（1）所有的叶结点都出现在最低的两层上。

（2）对任一结点，如果其右子树的高度为k，则其左子树的高度为k或k＋1。



完全二叉树



非完全二叉树

SHANGHAI JIAO TONG UNIVERSITY 上海交通大學

2 二叉树的重要特性

性质1 :二叉树第 i 层上至多有2^{i-1}个结点。

用归纳法证明:

归纳基: $i = 1$ 层时,只有一个根结点:

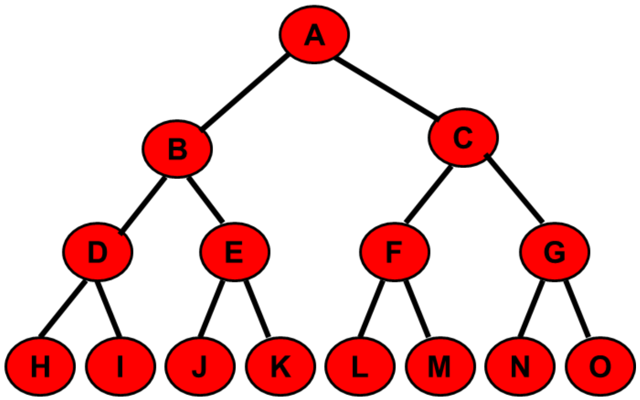
$2^{i-1} = 2^0 = 1$ 命题成立;

归纳假设: 假设 $i-1$ 命题成立,即:

第$i-1$层至多有结点$= 2^{i-1-1} = 2^{i-2}$ 个;

归纳证明: 二叉树每个结点至多有两棵子树,则

第i 层至多有结点 $= 2^{i-2}\times 2 = 2^{i-1}$ 个。



性质 2：

深度为 k 的二叉树上至多含 2^k-1 个结点(k≥1)。

证明:

基于性质1, 深度为 k 的二叉树上的结点总数的最大值是将二叉树每层上结点的最大值相加，所以深度为 k 的二叉树上含 结点数至多为:

$$2^0+2^1+\cdots\cdots+2^{k-1}=2^k-1$$

上海交通大學
SHANGHAI JIAO TONG UNIVERSITY

性质 3：

对任何一棵二叉树,若它含有n_0个叶子结点、n_2个度为 2 的结点，则必存在关系式:$n_0 = n_2+1$。

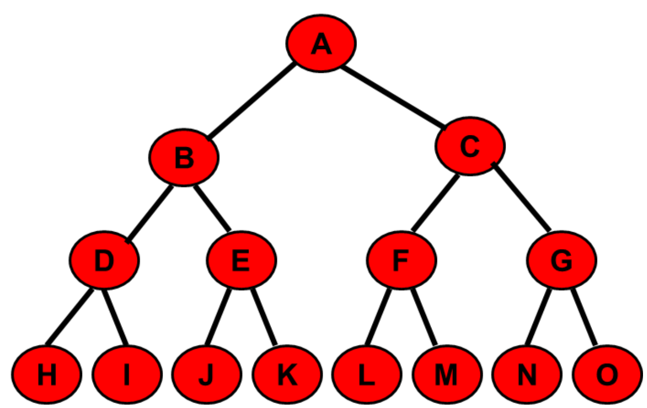
证明:

设 二叉树上结点总数： $n = n_0 + n_1 + n_2$

再根据树的性质： $b = n-1 = n_0 + n_1 + n_2 - 1$

由有二叉树分支总数： $b = n_1+2n_2$

两式右边相等得： $n_0 = n_2 + 1$ 。



性质 4：

具有 n 个结点的完全二叉树的深度为 $\lfloor \log_2 n \rfloor + 1$ 。

证明: 设完全二叉树的深度为 k

则根据性质2得 $2^{k-1} - 1 < n \leq 2^k - 1$

则 $2^{k-1} \leq n < 2^k$

即 $k-1 \leq \log_2 n < k$

k 只能是整数,因此: $k = \lfloor \log_2 n \rfloor + 1$ 。

性质 5：

若对含 n 个结点的完全二叉树从上到下且从左至右进行 1 至 n 的编号,则对完全二叉树中任意一个编号为 i 的结点:

(1) 若 $i=1$,则该结点是二叉树的根,无双亲,
否则,编号为 $\lfloor i/2 \rfloor$的结点为其双亲结点;

(2) 若 $2i \le n$则编号为 $2i$ 的结点为其左孩子结点,
否则,该结点无左孩子;

(3) 若 $2i+1 \le n$则编号为$2i+1$ 的结点为其右孩子,
否则,该结点无右孩子。

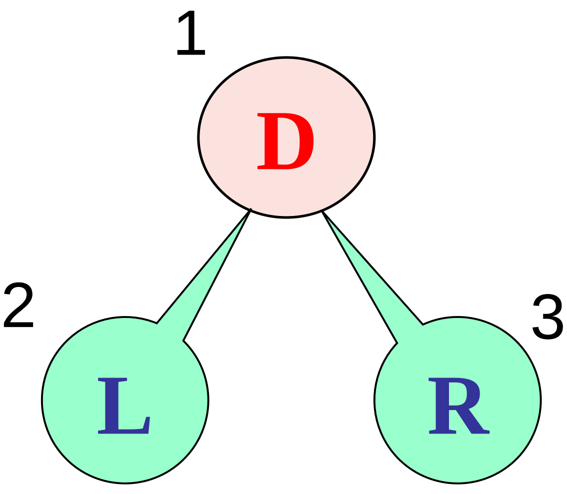
SHANGHAI JIAO TONG UNIVERSITY 上海交通大學

二叉树的遍历

- 二叉树的遍历讨论的是如何访问到树上的每一个结点
- 在线性表中，可以沿着后继链访问到所有结点。而二叉树是有分叉的，因此在分叉处必须确定下一个要访问的节点：是根结点、左结点还是右结点
- 根据不同的选择，有三种遍历的方法：**前序、中序和后序**
- 还有一种遍历方法是层次遍历

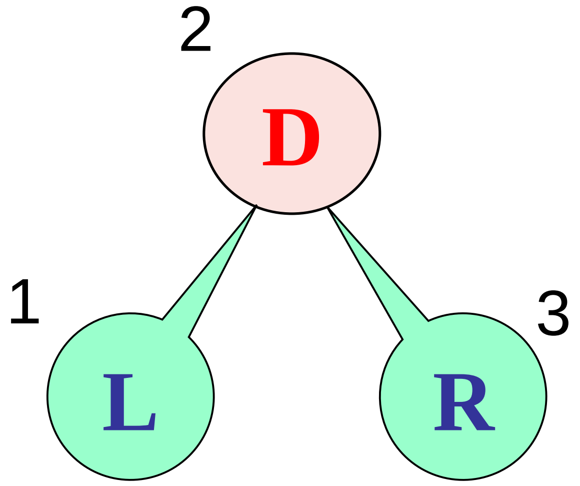
前序遍历

- 如果二叉树为空，则操作为空
- 否则
  - 访问根结点
  - 前序遍历左子树
  - 前序遍历右子树



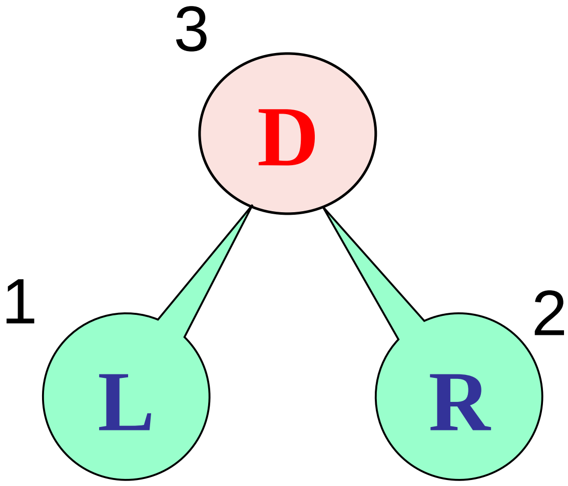
中序遍历

- 如果二叉树为空，则操作为空
- 否则
  - 中序遍历左子树
  - 访问根结点
  - 中序遍历右子树



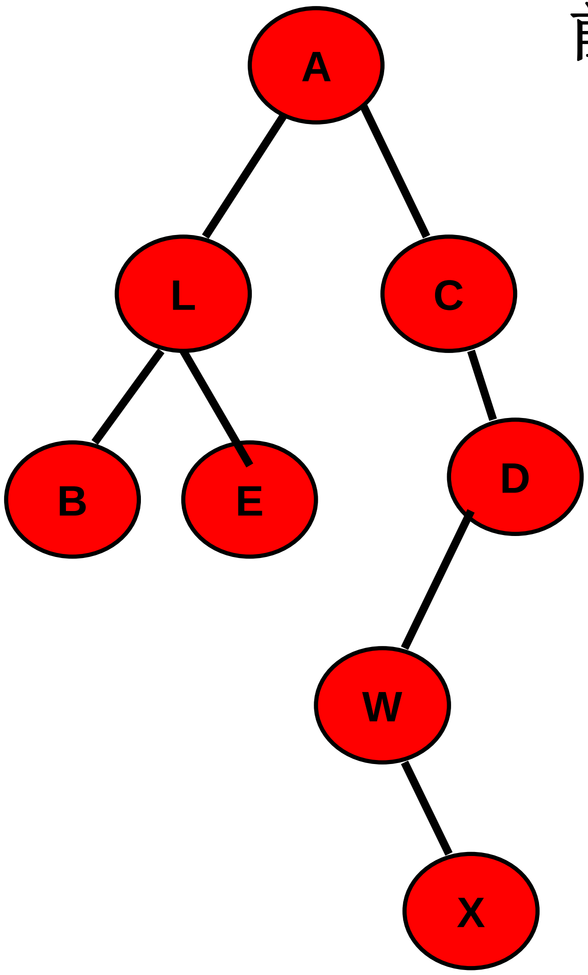
后序遍历

- 如果二叉树为空，则操作为空
- 否则
  - 后序遍历左子树
  - 后序遍历右子树
  - 访问根结点



层次遍历

- 先访问根结点，然后按从左到右的次序访问第二层的结点。在访问了第k层的所有结点后，再按从左到右的次序访问第k+1层。以此类推，直到最后一层。



前序： A、L、B、E、 C、D、W、X

中序：B、L、E、A、 C、W、X、D

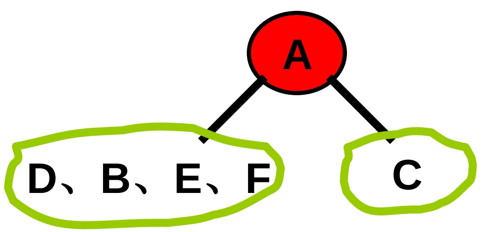
后序：B、E、L、X、 W、D、C、A

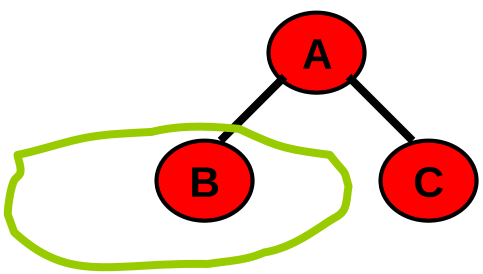
层次：A、L、C、B、E、D、W、X

前序 ＋ 中序 唯一确定一棵二叉树

前序: A、B、D、E、F、C

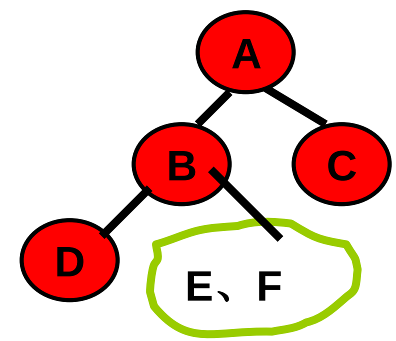
中序: D、B、E、F、A、C





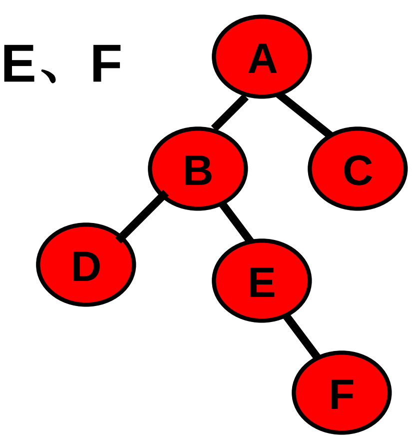
前序: B、D、E、F

中序: D、B、E、F



前序: E、F

中序: E、F



1. 已知一棵二叉树的前序和中序序列，求该二叉树的后序序列

前序序列：A, B, C, D, E, F, G, H, I, J

中序序列：C, B, A, E, F, D, I, H, J, G

后序序列：C, B, F, E, I, J, H, G, D, A

2. 设结点X和Y是二叉树中任意的两个结点，在该二叉树的先序遍历序列中 X 在 Y 之前，而在其后序遍历序列中 X 在 Y 之后，则 X 和 Y 的关系是（）

A. X 是 Y 的左兄弟

B. X 是 Y 的右兄弟

C. X 是 Y 的祖先

D. X 是 Y 的后裔

答案： C

二叉树的存储结构

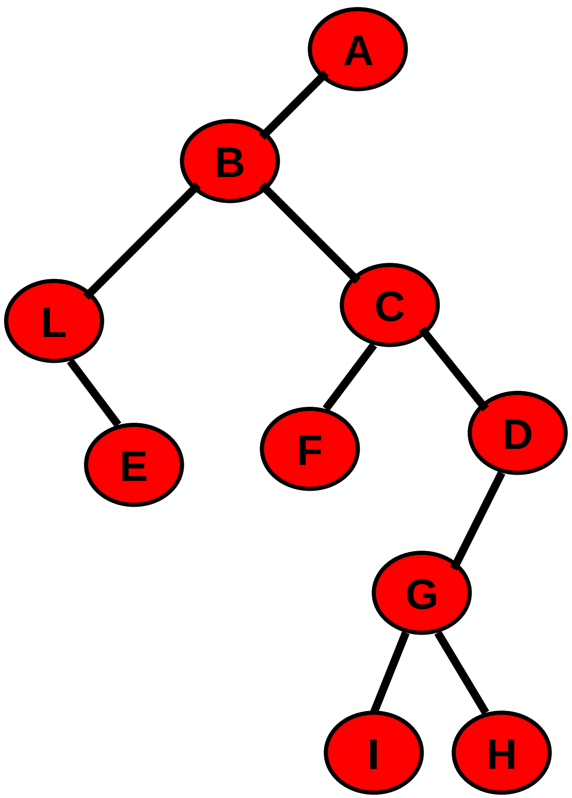
二叉树是非线性结构，结点最多有两个后继。存储结构有两种：顺序存储结构和链式存储结构。

一、二叉树的顺序存储

可用一维数组存储: **bt[n]**

顺序: 完全二叉树中将编号**i**的结点存在**bt[i]**中。

链接存储结构



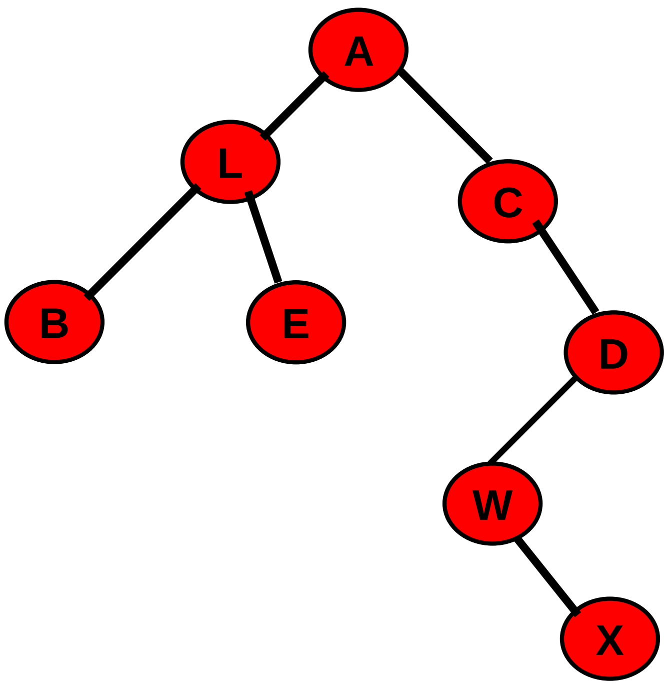
标准形式：（二叉链表）

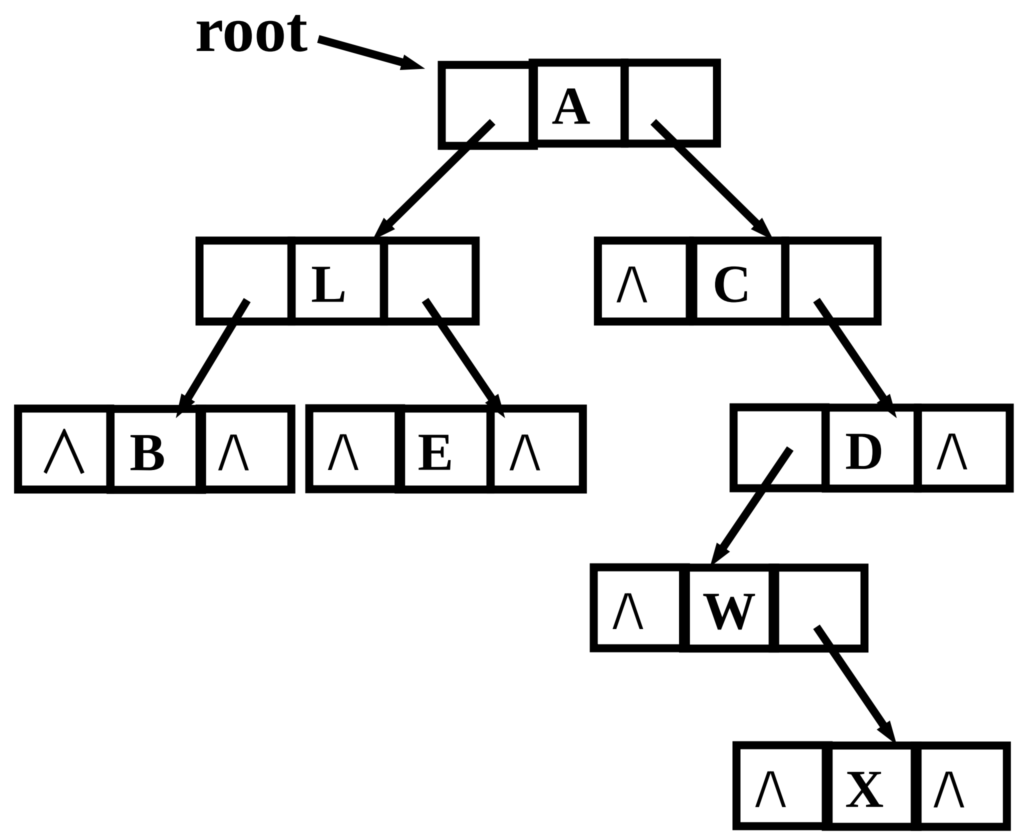
LEFT	DATA	RIGHT

广义标准形式：（三叉链表）

LEFT	DATA	PARENT	RIGHT

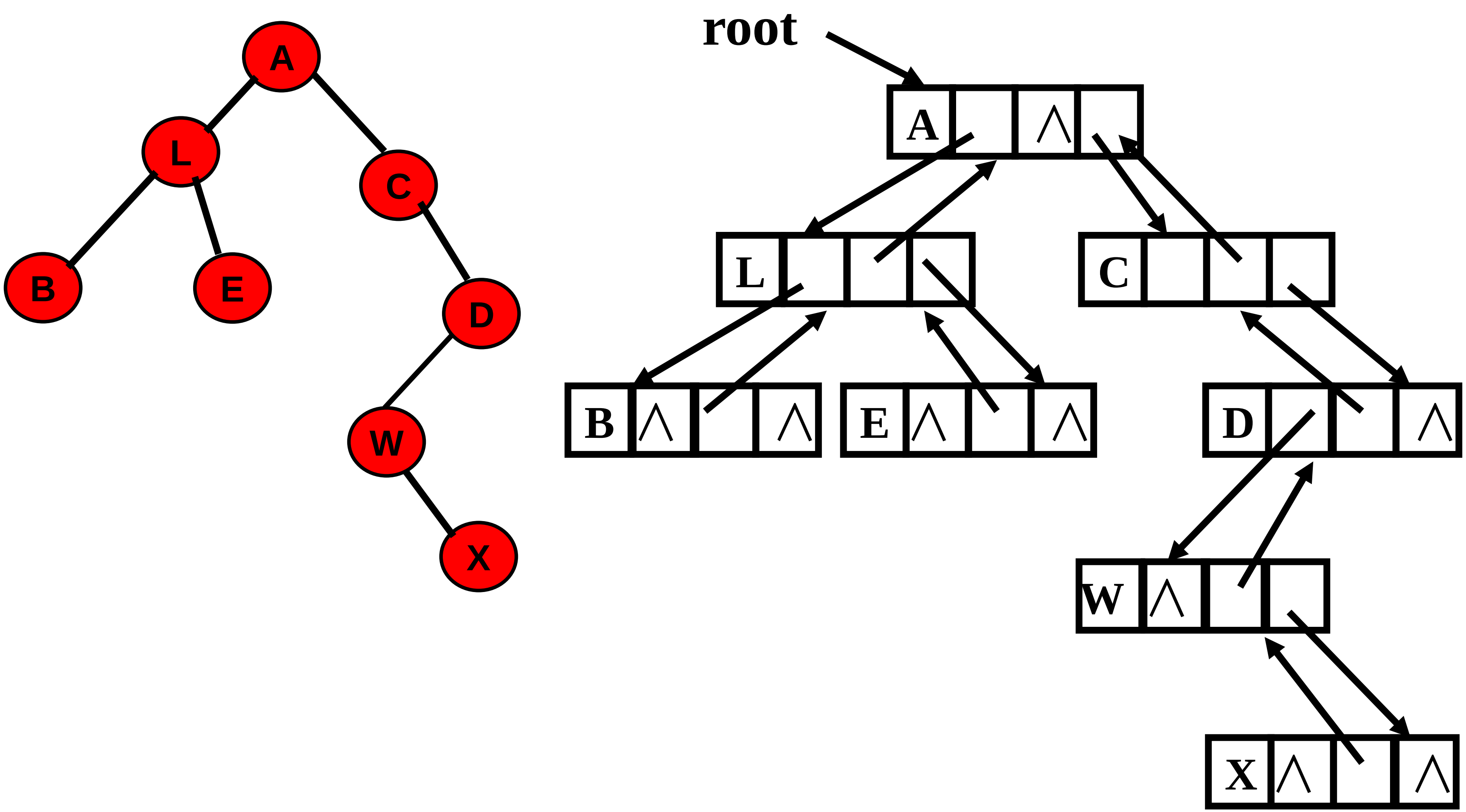
标准形式的实例





上海交通大学 SHANGHAI JIAO TONG UNIVERSITY

广义标准形式的实例



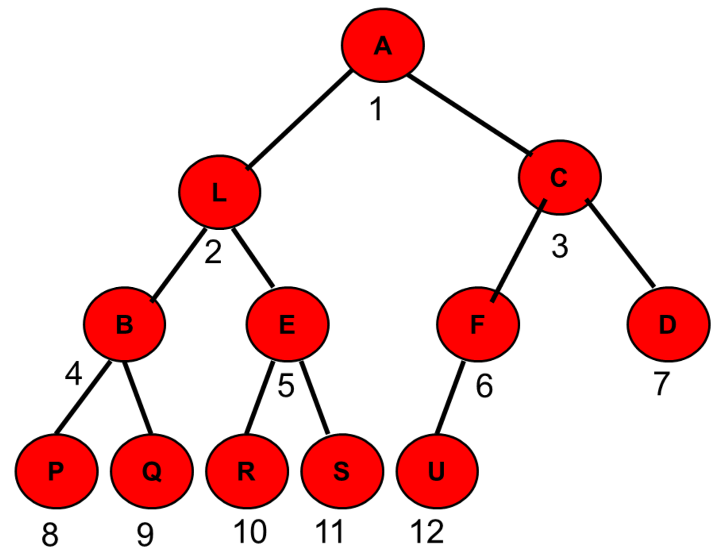
层次遍历

链接队列

```
template<class T>
void binaryTree<T>::levelOrder() const
{
    linkQueue< Node * > que;
    Node *tmp;

    cout << "\n层次遍历：";
    que.enQueue(root);

    while (!que.isEmpty()) {
        tmp = que.deQueue();
        cout << tmp->data << ' ';
        if (tmp->left) que.enQueue(tmp->left);
        if (tmp->right) que.enQueue(tmp->right);
    }
}
```

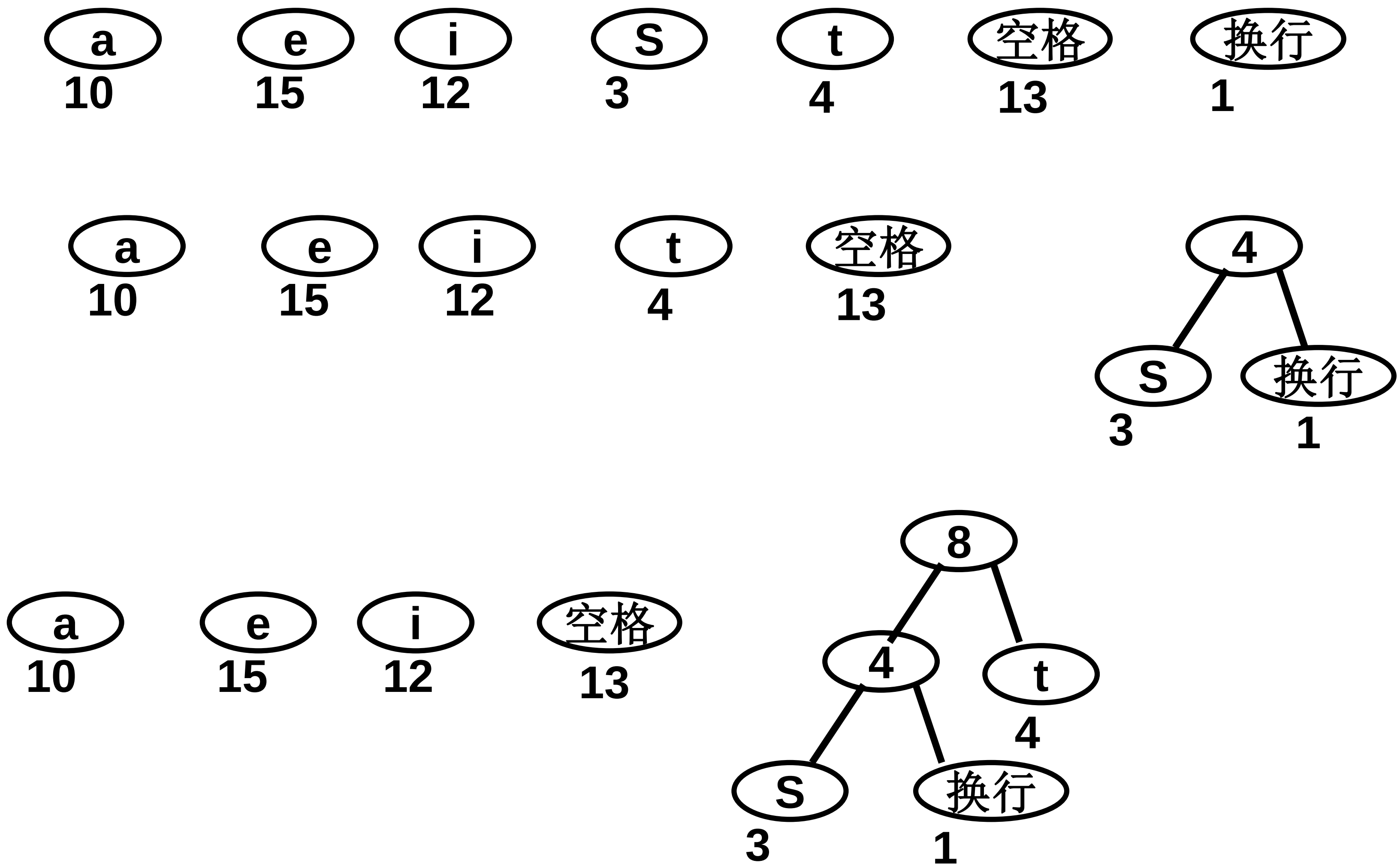


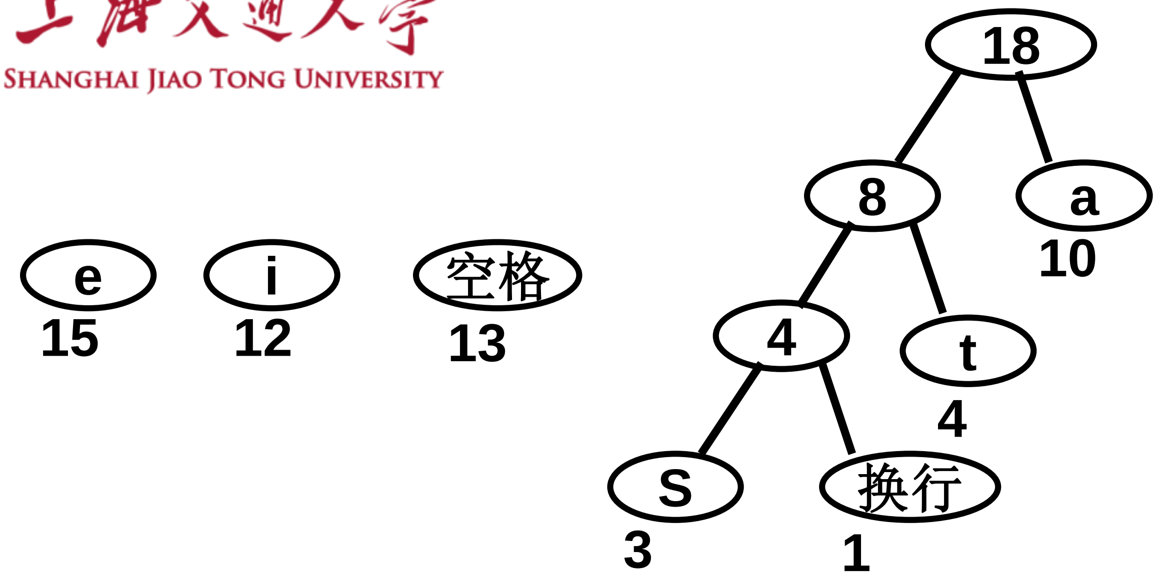
哈夫曼树与哈夫曼编码

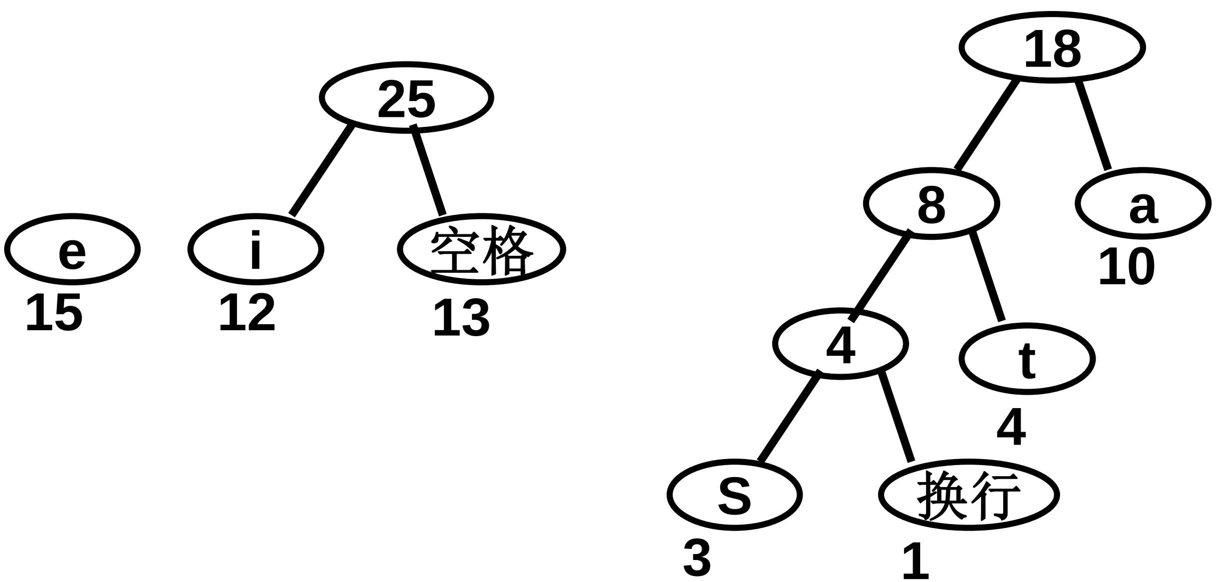
哈夫曼树

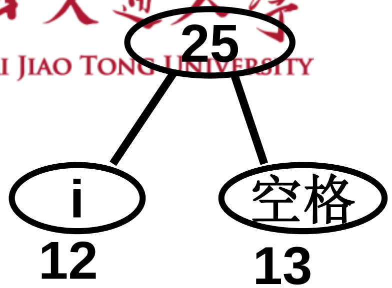
- 哈夫曼树是一棵最小代价的二叉树，在这棵树上，所有的字符都包含在叶结点上。
- 要使得整棵树的代价最小，显然权值大的叶子应当尽量靠近树根，权值小的叶子可以适当离树根远一些。

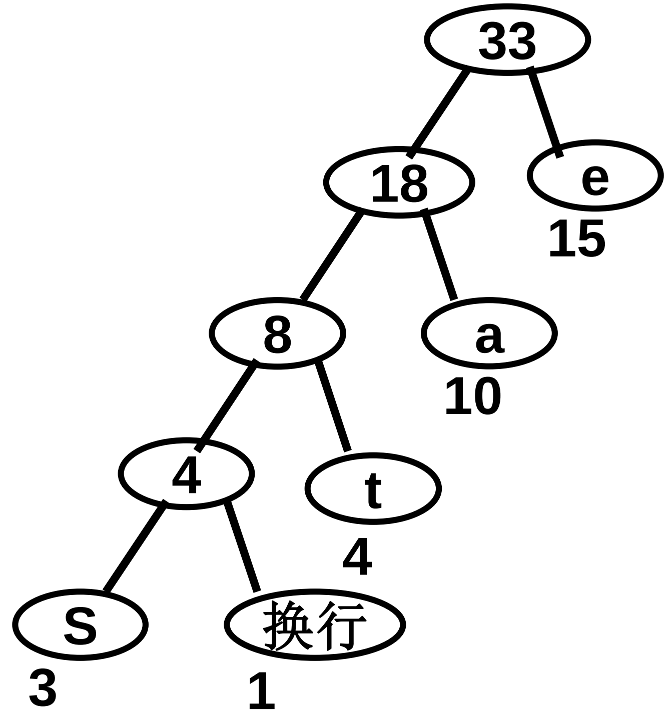
a(10), e(15), i(12), s(3), t(4), 空格(13), 换行(1)。

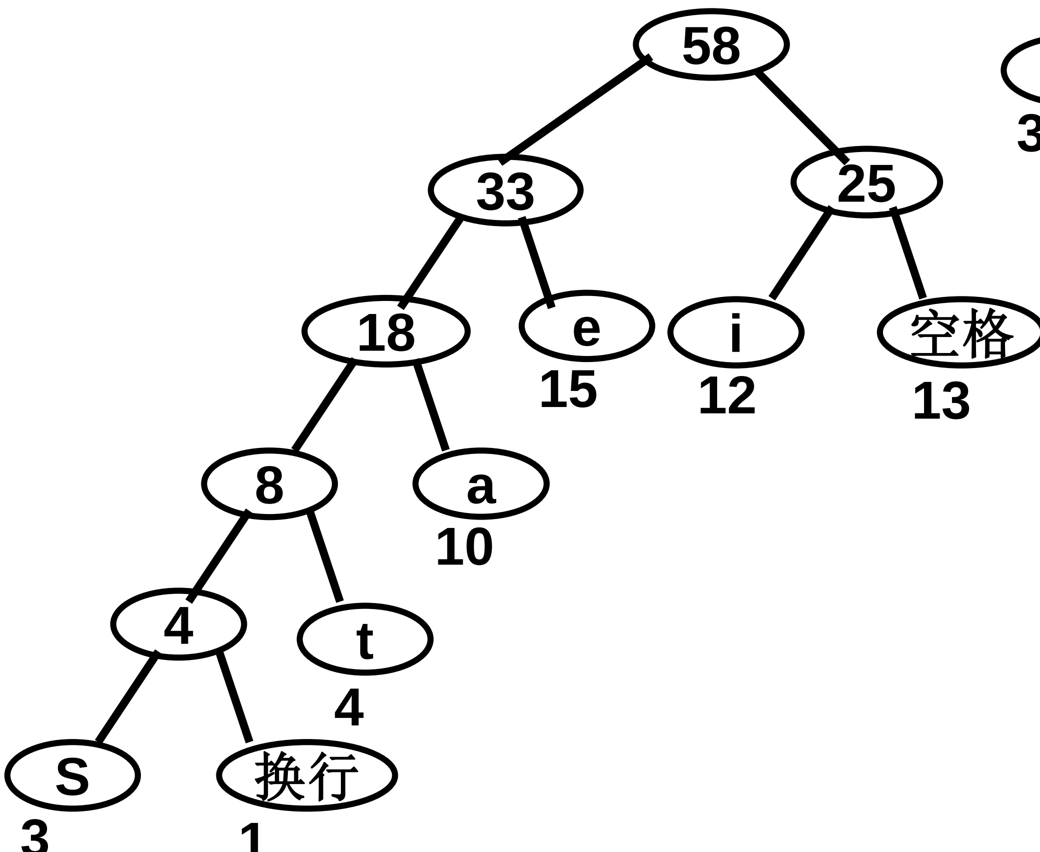



e
15
i
12
空格
13
18
8
a
10
4
t
4
S
3
换行
1


25
e
15
i
12
空格
13
18
8
a
10
4
t
4
S
3
换行
1


25
i
12
空格
13

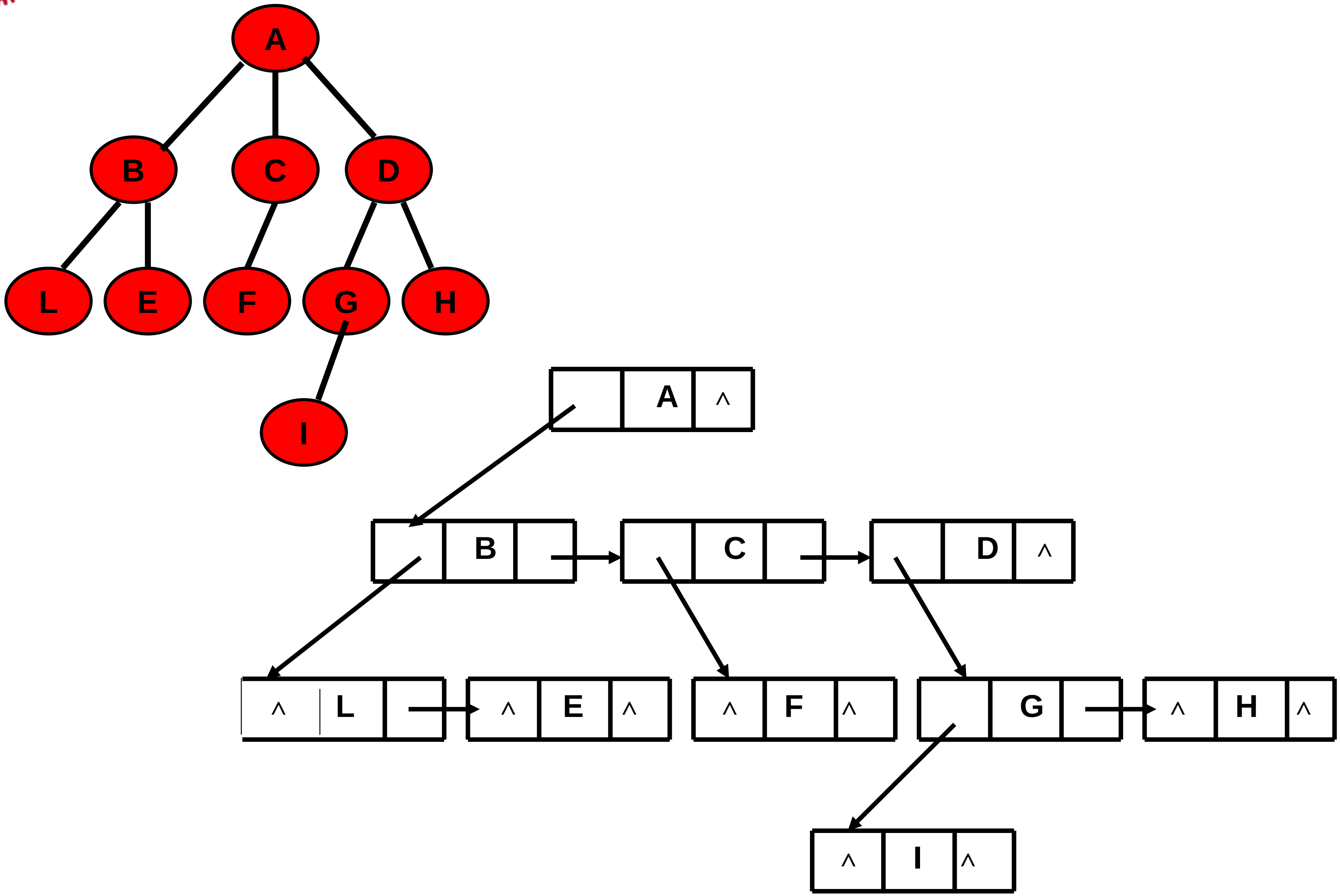

33
18
e
15
8
a
10
4
t
4
S
3
换行
1


58
33
25
18
e
15
i
12
空格
13
8
a
10
4
t
4
S
3
换行
1

孩子兄弟链表示法

实质上是用二叉树表示一棵树。树中的每个结点有数据字段、指向它的第一棵子树树根的指针字段、指向它的兄弟结点的指针字段。

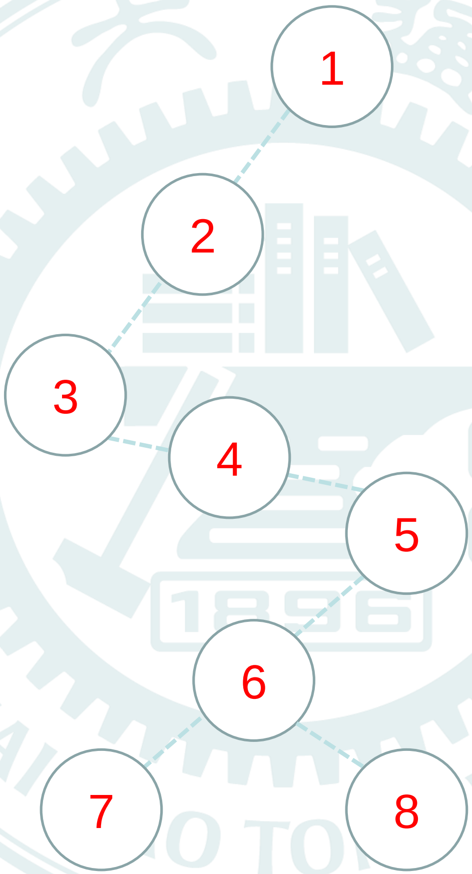
firstson	**data**	**nextsibling**


A
B
C
D
L
E
F
G
H
I
∧ A ∧
B
C
D ∧
∧ L
∧ E ∧
∧ F ∧
G
∧ H ∧
∧ I ∧

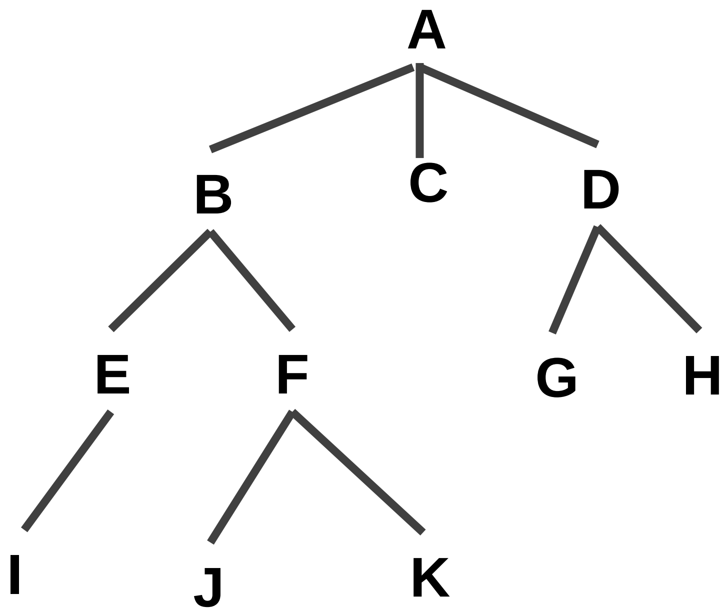
1. 已知字母使用频率为下表，求这8个字符的哈夫曼编码值。要求写出哈夫曼树的建立过程，约定在子树合并时，根的权值小的子树为左子树，大的为右子树。

字母	C1	C2	C3	C4	C5	C6	C7	C8
使用频率（%）	5	25	3	6	9	12	36	4

2. 下图是一棵树用左儿子、右兄弟法表示所对应的二叉树，请画出对应原树。



3. 下面的树按**左孩子，右兄弟**表示法转换成二叉树后有__3____个叶子节点



4. 所谓二叉树的层次遍历，是指从二叉树的第一层开始，按照从上至下，从左至右的顺序对结点逐个访问。设二叉树用标准二叉链表进行存储，试编写按层次遍历二叉树的程序。

```
template<class Type>class BinaryTree{
private:
    struct Node{    //二叉树的结点表示
        Type data;  //结点的数据场
        Node *left; //给出结点的左孩子的地址
        Node *right;//给出结点的右孩子的地址
    };
    Node *root;
public:
    BinaryTree():root(NULL) //二叉树的构造函数
    ~BinaryTree(){clear();} //二叉树的析构函数
    void layorder(Node<Type>*&t)const
};
```

参考答案：借用队列来完成，只要队列不为空，就出队。然后判断结点是否有左孩子和右孩子，如有就依次输出左、右孩子的值，让左、右孩子进队。

```
template<class Type>void BinaryTree::
layorder(Node *t)const
{
  initqueue(q);     //队列初始化
  if(T!=NULL){
    cout<<T->data<<endl;
    enqueue(q,T);   //入队
    while(!emptyqueue(q)){  //若队列非空
        outqueue(q,p);
        if(p->lchild!=NULL){
            cout<<p->lchild->data<<endl;
            enqueue(q,p->lchild);  //入队
        }
            if(p->rchild!=NULL){
            cout<<p->rchild->data<<endl;
            enqueue(q,p->rchild);  //入队
        } }}}
```

5. 假定在一棵二叉树中,度为2的结点数为15,度为1的结点数为30,则叶子结点数为( )个。
A. 15
B. 16
C. 17
D. 47
参考答案：B

6. 二叉树的深度为k,则二叉树最多有( )个结点。
A. 2k
B. 2^(k-1)
C. (2^k) - 1
D. 2k-1

参考答案：C

7. 设a,b为一棵二叉树上的两个结点,在中序遍历时, a在b前面的条件是( )。
A. a在b的右方
B. a在b的左方
C. a是b的祖先
D. a是b的子孙
参考答案：B

8. 将一棵有100个结点的完全二叉树从根这一层开始,每一层上从左到右依次对结点进行·编号,根结点的编号为1,则编号为49的结点的左孩子编号为( ).
A. 98
B. 99
C. 50
D. 48
参考答案：A

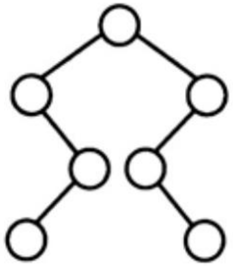
9.具有65个结点的完全二叉树其深度为( )（根的深度为1）
A. 5
B. 6
C. 7
D. 8
参考答案：C

10.已知一棵完全二叉树的第6层（设根为第1层）有8个叶结点，则该完全二叉树的结点个数最多是：()
A.119
B.52
C.111
D.39

参考答案：C

11. 要使一棵非空二叉树的先序序列与中序序列相同，其所有非叶结点须满足的条件是
A．只有左子树
B．只有右子树
C．结点的度均为1
D．结点的度均为2
参考答案：B

12．已知一棵二叉树的树形如下图所示，其后序序列为e，a，c，b，d，g，f，树中与结点a
同层的结点是



A．c　B．d　C．f　D．g
参考答案：B

13.已知字符集{a，b，c，d，e，f，g，h}，若各字符的哈夫曼编码依次是0100，10，0000，0101，001，011，11，0001，则编码序列 0100011001001011110101的译码结果是( )

A．a c g a b f h　　B．a d b a g b b
C．a f b e a g d　　D．a f e e f g d

参考答案：D

14．若森林F有15条边、25个结点，则F包含树的个数是()
A．8　　B．9　　C．10　　D．11

参考答案：C

15．先序序列为a，b，c，d的不同二叉树的个数是()
A．13　　B．14　　C．15　　D．16
参考答案：B