

ROOT



树形结构1 树的概念

竞入门课程

本课学习目录：

第一节：树的概念

第二节：二叉树的概念

第三节：二叉树的遍历

第四节：多叉树转二叉树

重庆南开信竞入门课程

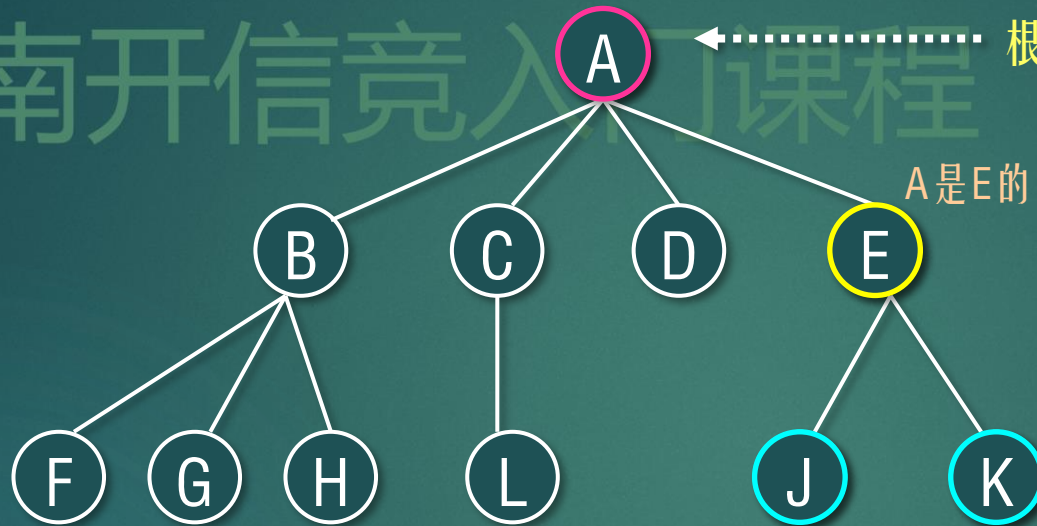
重庆南开信竞入门课程

第一节：树的概念

重庆南开信竞入门课程

树

树是由 $n(n \geq 0)$ 个节点和 $n-1$ 条边构成集合。特点是任何两个节点间有且仅有一条路径。



根节点：没有父亲的节点。
一棵树只有一个根节点。

A是E的父亲节点

J和K是E的儿子节点
E是J和K的父亲节点

包含一个节点的所有子孙和该节点本身的集合，称为子树

每个节点可有0个或多个儿子节点

除根节点外，每个节点有且只有一个父亲节点

没有儿子的节点称为叶节点，比如 F G H L J K 都是叶节点

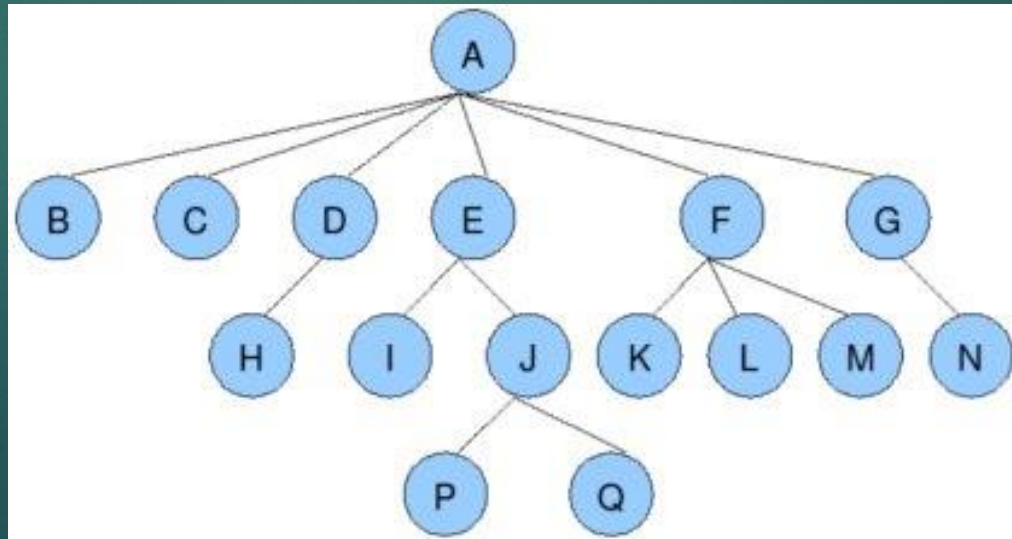
父亲相同的节点称为兄弟节点，比如 F G H 是兄弟

节点的儿子的个数称为节点的度，比如 A 的度数为4

祖先、子孙

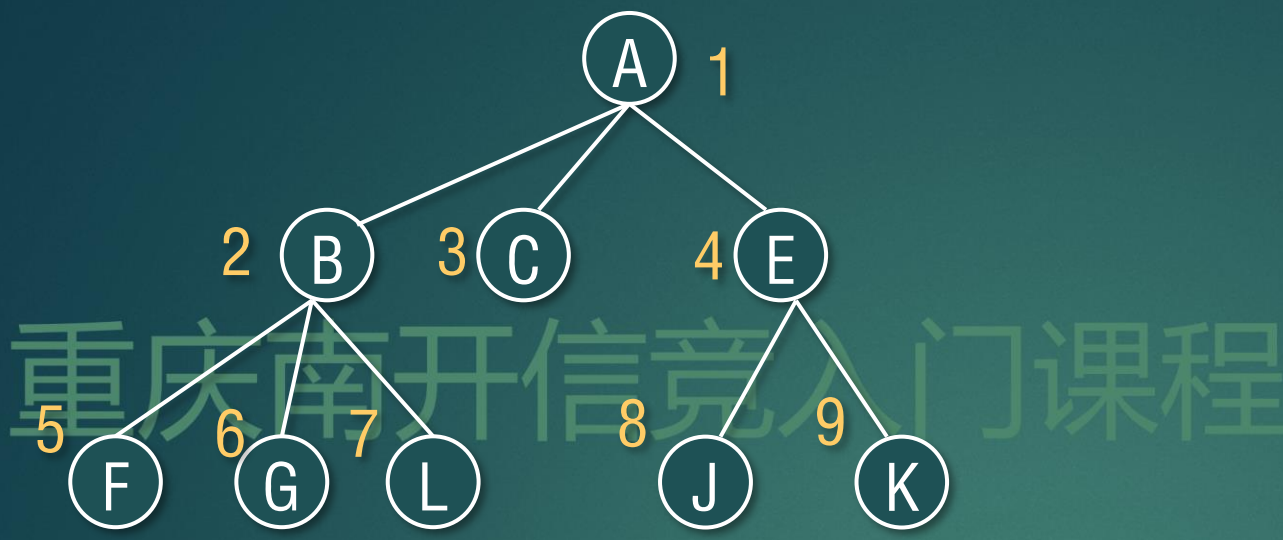
树的概念大集合：

- (1) 空树与非空树：空树它没有任何的节点；非空树，至少含有一个节点。
- (2) 根 (Root)：非空树中有且仅有一个节点没有父亲节点，那个节点就是根。
- (3) 子树 (Subtree)：在一棵非空树中，除根外，其余所有节点可以分为多个互不相交的集合。每个集合本身又是一棵树，称为根的子树。
- (4) 节点 (Node)：表示树中的元素。
- (5) 节点的度 (Degree)：一个节点拥有的子树(子节点)数目称为该节点的度。
- (6) 叶子节点 (Leaf)：度为0的节点。
- (7) 孩子 (Child)：节点的下一层节点 (节点子树的根称为该节点的孩子)。
- (8) 父亲 (Parents)：孩子节点的上层节点叫该节点的父亲。
- (9) 兄弟 (Sibling)：拥有共同父亲的节点的节点。
- (10) 树的度(叉)：一棵树中最大的节点度数。
- (11) 深度 (Depth)：树中节点所在的层数。
- (12) 森林 (Forest)：是多棵互不相交的树的集合。
- (13) 祖先：是指从i号节点到根节点的路径之中的所有的节点，都叫i节点的祖先；
- (14) 子孙：以i号节点为根的子树中的所有节点，都是i号点的子孙；



树的存储

一般用数组来存储树形结构，
用数组下标来表示节点的位置



数组下标

1 2 3 4 5 6 7 8 9

节点

A	B	C	E	F	G	L	J	K
---	---	---	---	---	---	---	---	---

父亲

0	1	1	1	2	2	2	4	4
---	---	---	---	---	---	---	---	---

儿子

2	5	8
3	6	9
4	7	

多叉树的缺点：

不方便存储，需要 $n \times n$ 的矩阵
不方便操作

重庆南开信竞入门课程

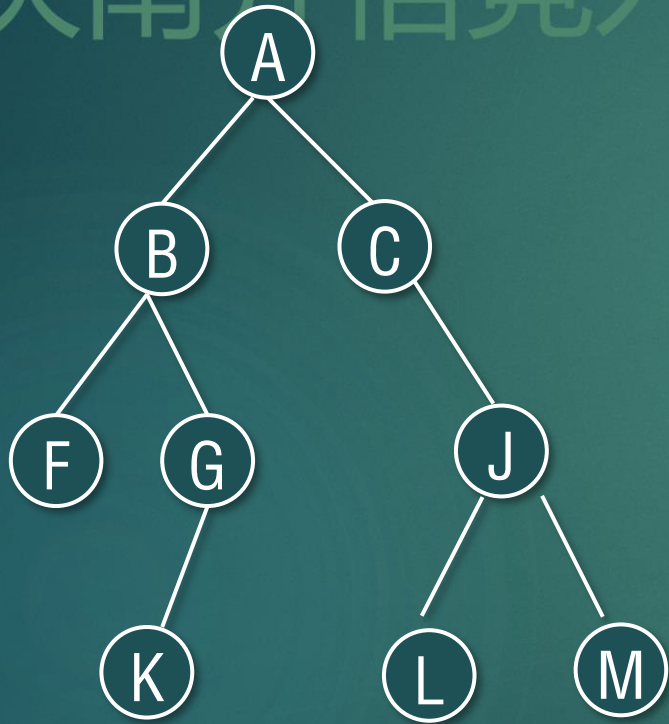
第二节：二叉树的概念

重庆南开信竞入门课程

二叉树 (Binary tree)

二叉树就是每个节点最多只有两个子节点的树。

重庆南开信竞入门课程



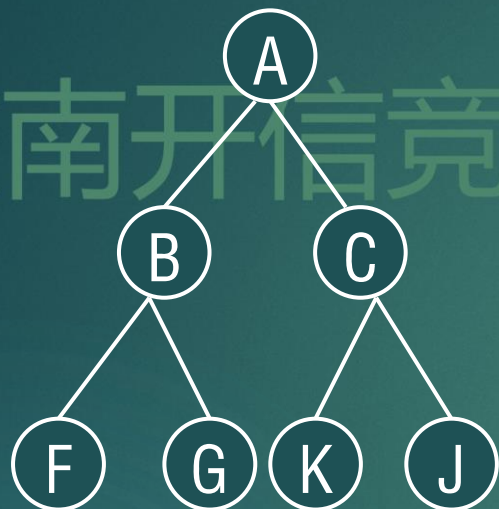
二叉树的特点：

1. 第 i 层上最多有 2^{i-1} 个节点
2. 高度为 h 的二叉树最多有 $2^h - 1$ 个节点
3. 在非空二叉树中，
叶节点的个数等于度为2的节点个数加1

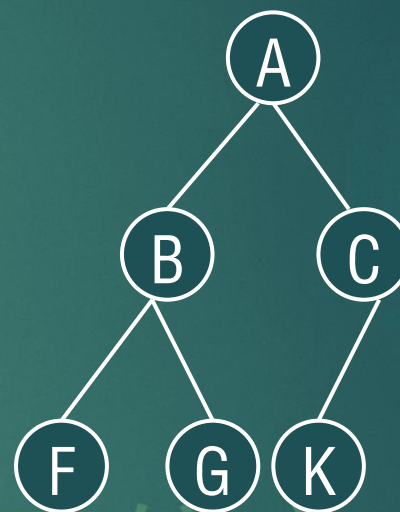
重庆南开信竞入门课程

二叉树

每层节点数都达到最大的二叉树叫 **满二叉树**



满二叉树



完全二叉树

除最底层外，其余各层节点都达到最大值，且最底层的节点集中在 **左侧** 的连续位置上的二叉树叫 **完全二叉树**。

二叉树的存储

一般用多维数组或结构体类型数组来存储



1	2	3	4	5	6
A	B	C	F	G	J
0	1	1	2	2	3
2	4	0	0	0	0
3	5	6	0	0	0

```
struct node{  
    char data;  
    int father,left,right;  
};  
node tree[100];
```

```
tree[2].data='B';  
tree[2].father=1;  
tree[2].left=4;  
tree[2].right=5;
```

重庆南开信竞入门课程

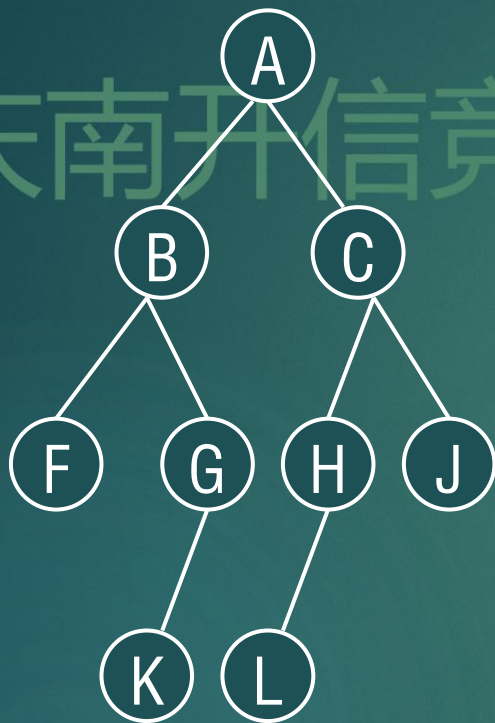
第三节：二叉树的遍历

重庆南开信竞入门课程

二叉树的遍历 Traversal

NK
OI

以固定的顺序访问二叉树各个节点，每个节点均恰好被访问一次。



左边二叉树遍历的顺序可以是：A B C F G H J K L

这种遍历称为**层次遍历**

二叉树按递归遍历有**前序**、**中序**、**后续**三种遍历

前序遍历：先访问**根**节点，然后访问**左**子树，最后访问**右**子树。

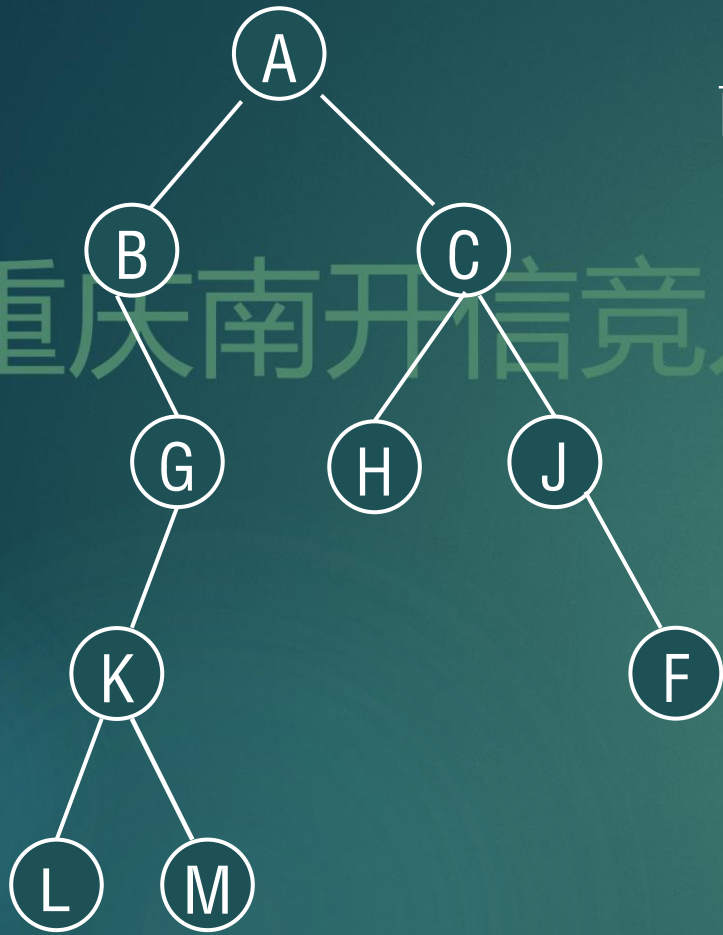
A B F G K C H L J

中序遍历：先访问**左**子树，然后访问**根**节点，最后访问**右**子树。

F B K G A L H C J

后序遍历：先访问**左**子树，然后访问**右**子树，最后访问**根**节点。

F K G B L H J C A



前序 A B G K L M C H J F

中序 B L K M G A H C J F

后序 L M K G B H F J C A

重庆南开信竞入门课程

已知一棵二叉树的中序和后序遍历的顺序，请画出这棵二叉树

中序：K,B,V,G,M,J,H,F,X

后序：K,V,M,G,B,H,X,F,J



已知“中序和前序”或者“中序和后序”可以推出这棵二叉树

二叉树遍历的递归实现

NK
OI

前序

```
void qian(int p)
{
    if(p>0)
    {
        cout<<tree[p].data;
        qian(tree[p].left);
        qian(tree[p].right);
    }
}
```

中序

```
void zhong(int p)
{
    if(p>0)
    {
        zhong(tree[p].left);
        cout<<tree[p].data;
        zhong(tree[p].right);
    }
}
```

后序

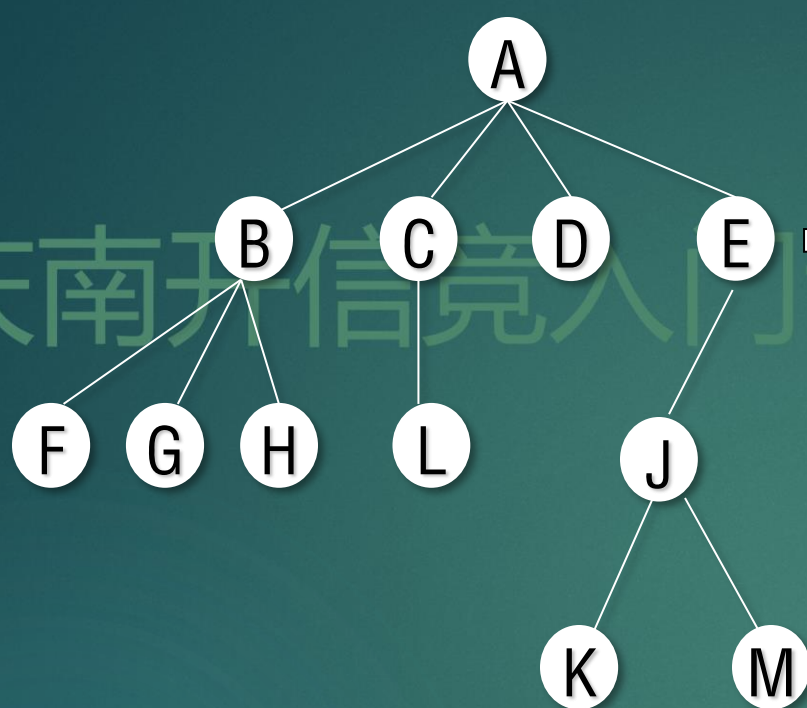
```
void hou(int p)
{
    if(p>0)
    {
        hou(tree[p].left);
        hou(tree[p].right);
        cout<<tree[p].data;
    }
}
```

遍历时间复杂度 $O(n)$

重庆南开信竞入门课程

第四节：多叉树转二叉树

重庆南开信竞入门课程



兄弟关系

兄弟关系

父子关系

多叉树转二叉树的规则：每个节点的左起第一个孩子变成它的左孩子，右边第一个兄弟变成它的右孩子。

多叉树转二叉树，代码

重庆南开信竞入门课程

```
cin >> n;           //树中共n个节点
for (i=1; i<n; i++)  //读入n-1条边
{
    cin >> x >> y;    //x是y的父亲
    right[y]=left[x]; //x的左儿子是y的兄弟，成为y的右儿子
    left[x]=y;        //y成为x新的左儿子
}
```

重庆南开信竞入门课程

老儿子变成新儿子的右儿子
新儿子成为父亲的左儿子

重庆南开信竞入门课程

课后习题讲评

重庆南开信竞入门课程

遍历二叉树1 NKOJ4376

题面描述:

给出一棵n个节点(编号1到n)的二叉树, 输出该二叉树的前序、中序、后序遍历序列。

输入格式

第一行, 一个整数n

接下来n-1行, 每行两个整数x和y, 表示x是y的父亲

(为保证答案唯一, 我们规定, 根据输入顺序, 若y是x的第一个出现的儿子, y为x的左儿子, 否则为右儿子)

输出格式

三行, 每行n个空格间隔的整数, 第一行表示前序遍历序列, 第二行表示中序遍历的序列, 第三行表示后序遍历的序列

$1 \leq n \leq 10000$

重庆南开信竞入门课程


```
int n,Left[10005],Right[10005]; //Left[i]记录i号点的左儿子编号
bool mark[100005]; //mark[i]标记i号点是否有父亲
```

NK
OI

```
int main()
{
    int i,x,y,Root;
    cin>>n;
    for(i=1;i<n;i++)
    {
        cin>>x>>y; //x是y的父亲
        if(Left[x]==0) Left[x]=y;
        else Right[x]=y;
        mark[y]=true;
    }
    for(i=1;i<=n;i++) //找出根节点
        if(mark[i]==false){Root=i; break;}
    Qian(Root); cout<<endl;
    Zhong(Root); cout<<endl;
    Hou(Root); cout<<endl;
}
```

重庆南开信竞入门课程

遍历二叉树2 NKOJ3909

题面描述：

给出两个由大写字母构成的字符串(长度不超过26)，一个表示二叉树的前序遍历序列，一个表示二叉树的中序遍历序列，请你计算出该二叉树的后序遍历序列。

样例输入

ABGKLMCHJF
BLKMGAHCJF

样例输出

LMKGBHFJCA

重庆南开信竞入门课程

```
string Qian,Zhong;
int len;
void FindHou(int L,int R)    //当前讨论中序序列的[L,R]区间
{
    if(L>R) return;
    for(i=0;i<len;i++)
        for(j=L;j<=R;j++)
            if(Qian[i]==Zhong[j])
            {
                FindHou(L,j-1);
                FindHou(j+1,R);
                cout<<Zhong[j];
                return;
            }
}

int main()
{
    cin>>Qian>>Zhong;
    len=Qian.length();
    FindHou(0,len-1);
}
```

//Zhong[j]为[L,R]表示的子树的根，后续遍历该子树

重庆南开信竞入门课程