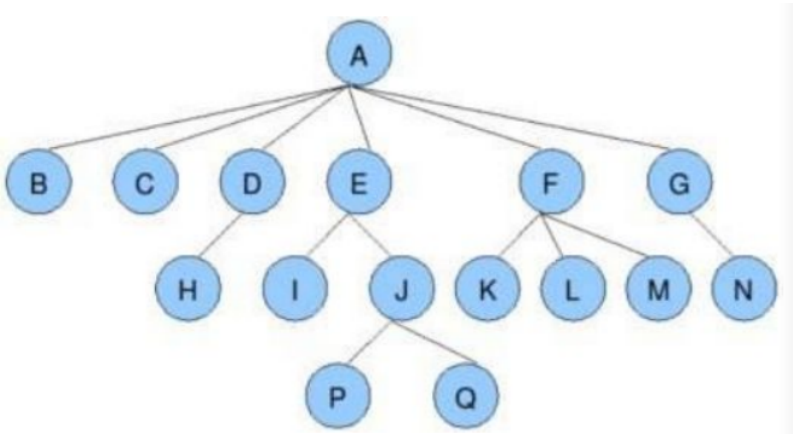
二叉树

一、二叉树的一些基本概念：



1、**节点的度：一个节点含有的子树个数称为该节点的度；如：A的为 6**

2、**树的度：一棵树中，最大的节点的度称为概述的度；如：树的度为6**

3、叶子节点(终端节点)：度为0的节点；如：B,C,H,I……等节点为叶子节点

4、双亲结点或父节点：若一个节点含有子节点，则整个结点称为其子节点的父节点；如：A 是B的双亲结点

5、孩子结点(子节点)：一个节点含有的跟节点称为该节点的孩子节点；如：B 是 A的孩子节点、

6、根节点：一棵树中，没有双亲结点的结点；如：A

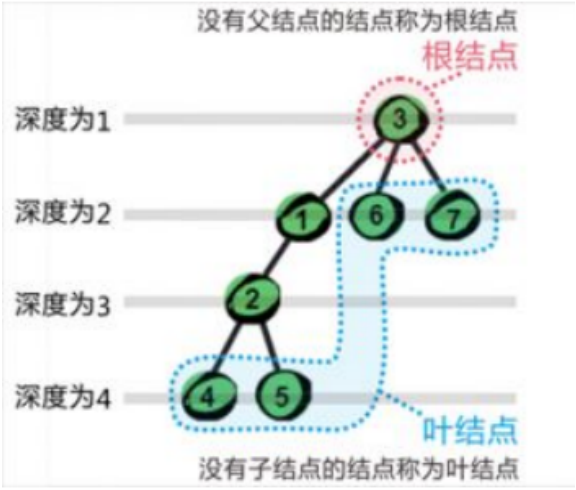
7、**节点的层数：从跟开始定义起，根为为第1曾，根的子节点为第二层，以此类推**

8、**树的高度(深度)：树中结点的最大层次；如：上树的高度为4；**

课外延申：

一个二叉树树的度为N，度为0的节点数为N0,度为1的节点数为N1，度为2的节点数为N2，N=N0+N1+N2;

其中N0 = N2+1;



9、树的形式：（了解，不常用）

```
class Node{
 int val; //树中存储的数据
 Node left; //第一个孩子引用
 Node next; //下一个兄弟的引用
}
```

二、二叉树

1、概念：

2、特点：

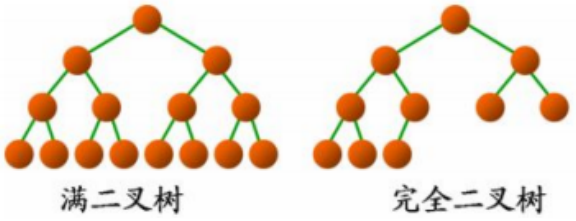
该树的每个节点最多有两棵子树(即每个节点的的不大于2)；

二叉树的子树有左右之分，且不能乱

3、特殊的二叉树：

（1）完全二叉树：完全二叉树是特殊的满二叉树

（2）满二叉树：每一层的节点数都达到最大值，则这个二叉树为满二叉树。若一个二叉树的层数为k，且节点总数是(2^k)-1。



4、二叉树的遍历——前中后序

三种遍历方法：把树分为三部分：根+左子树+右子树

（1）递归的思想：

```
private void preOrder(Node root){
 if(root==null){
 return;
 }
 System.out.println(root);
 preOrder(root.left);
 preOrfer(root.right);
}
```

（2）遍历的思想：

1）记录的位置独立于遍历过程之外；

2）每次都需要充值记录遍历；

```
class Solution{
 private List<Integer> list = new ArrayList<>();
```

```
3   private preOrder0(Node root){
4   //统计根的值
5   list.add(root);
6   preOrder0(root.left);
7   preOrder0(root.right);
8   }
9   public List<Integer> preOrder（Node root）{
10   list.clear();
11   preOrder0(root);
12   return list;
13   }
14  }
```

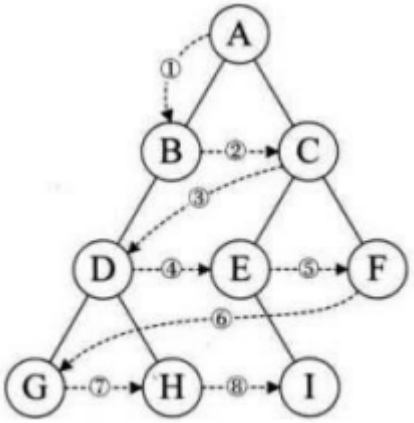
3）汇总的思想：

三部分：根 + 左子树 + 右子树

整棵树的结果 = 汇总（左子树，右子树）

```
1  class Solution{
2   public List<Integer> preOrder(Node root){
3   List<Integer> list = new ArrayList<>();
4   if(root==null){
5   return list
6   }
7   List<Integer> left = preOrder(root.left);
8   List<Integer> right = preOrder(root.right);
9
10   list.add(root);
11   list.addAll(left);
12   list.addAll(right);
13   return list;
14   }
15  }
```

5、二叉树的遍历——层序遍历：



设二叉树的根节点所在层数为1，层序遍历就是从所在二叉树的根节点出发，首先访问第一层的树根节点，然后从
左到右访问第2层上的节点，接着是第三层的节点，以此类推，自上而下，自左至右逐层访问树的结点的过程就是
层序遍历。

6、二叉树的表示：

```
class Node{
 int val;
 Node left;
 Node right;
 Node parent;
}
Node root = null;
```

练习：

1、二叉树的遍历（前中后、层序）各种方法；
2、检查两棵树是否相同；
3、检查A树是否为B树的子树
4、求二叉树的高度——只能用汇总思想
5、求二叉树的节点个数
6、求二叉树的叶子节点个数
7、求二叉树的第K层节点的个数
8、在一个二叉树中查找是否有某个节点
9、求二叉树的最大深度
10、对称二叉树
11、互为镜像二叉树
12、判断一个二叉树是否是平衡二叉树
13、二叉树的构建及遍历

14、二叉树的分层遍历
15、给定一个二叉树, 找到该树中两个指定节点的最近公共祖先
16、二叉树搜索树转换成排序双向链表
17、根据一棵树的前序遍历与中序遍历构造二叉树
18、根据一棵树的中序遍历与后序遍历构造二叉树
19、二叉树创建字符串