树

有根树

存在唯一一个$root$节点v_0，使得$\forall v \in A$,都有一条从v_0到v的路径。记为(T, v_0)

性质

1. T中不含任何环路。
2. v_0是T的唯一根节点。
3. 除根节点外，每个顶点的**入度**均为1， 而根节点的入度为0

层级表示法：根节点位于第0层，从上往下分别层数+1。

- 没有子节点的称为叶节点。
- 不是叶节点的称为内部节点
- 不是根节点的内部节点称为分支节点
- 树的最大层数即树的高度

家族关系:

- 祖先节点：父节点的所有上级节点
- 父节点：紧邻的上级节点
- 兄弟节点：由同一个父节点引出的同级节点
- 子节点：紧邻的下级节点
- 后代节点：子节点的所有下级节点

相关术语：

- **有序树**：每一层的顶点均存在序关系
- **n叉树**：任一顶点的子代数目不超过n
- **完全n叉树**：除叶节点外，所有顶点**均恰有**n个子代
- **二叉树**：$n = 2$的n叉树的特例

有根树的子树 令$T(v)$表示v及其内所有的边。则(T, v)构成一颗以v为根的树。 满足：连通性、唯一路径性、无环性。

有序二叉树 所有的N叉树都可以被转化为一颗有序二叉树

树的搜索

- **中序遍历**：左$\rightarrow$根$\rightarrow$右
- **先序遍历**：根$\rightarrow$左$\rightarrow$右
- **后序遍历**：左$\rightarrow$右$\rightarrow$根

特殊的树

- **位置树**：表示每个每个节点再树中的位置具有语义上的意义
- **霍夫曼编码树**：用01表示二叉树的左右枝。01串则可以表示具体节点

- **红黑树**：自平衡树（最长路径不会超过最短路径的两倍）

无向树

无向树是树的**对称闭包** 当(a,b)以及(b,a)都属于T时，集合a,b称为一个**无向边**，a和b称为相邻顶点。

基础概念

- **简单路径**：设p是对称关系R中的一条路径，p中任意两条边不对应同一条无向边。
- **简单环**：对于$p: v_1 v_2 \dots v_n$，若仅有$v_1 = v_n$，则p是一条简单环。
- **无环的**：若对称关系R中不包含任何简单环，则称R是**无环的**
- **连通的**：若对称关系R中的任意两个顶点之间都存在一条路径，则称R是连通的。

无向树的性质

- 设R是集合A上的一个对称关系。**当且仅当**R是**连通**的且无环**无环**的，R才是一颗无向树。
- 如果T是一颗无向树，那么对于任意的两个顶点u和v，在T中都存在一条**唯一**的从u到v的路径。
- 设T是一颗无向树，e是T中任意一条边，则$T-e$不再是**连通的**。
- 设T是一颗无向树，u和v是T中两个不相邻的顶点，则在$T+(u,v)$中必定包含一个环。

其他知识

边和节点的数量：一颗包含n个节点的树恰有$n-1$条边 **生成树**：如果R是集合A上的一个**对称、连通**关系，而T是一颗树，并且和R具有完全相同的顶点，则A上的树T是R的生成树。

最小生成树

连通无向图的一个生成树，**包含所有顶点、无环、最小总权重。** 构造方法：

- **普利姆算法**：通过局部最优实现全局最优。每次找权重最小的边。
- **克鲁斯卡尔算法**：找图中最小的边，但是不成环。