

树的基本概念

回顾

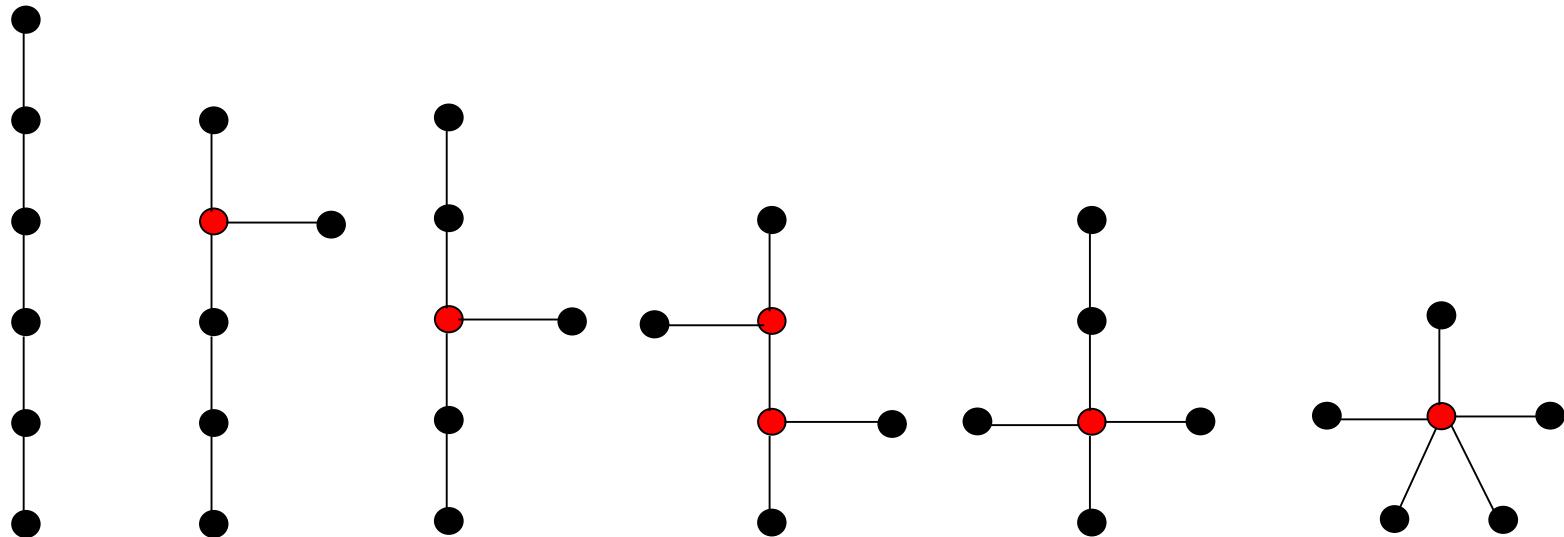
- 问题1：什么是二部图及其匹配？
 - 两个无内部边的顶点集；互不相邻的边的集合
- 问题2：二部图中的有哪些匹配？
 - 完备匹配：一个部饱和，充要条件为Hall定理
 - 最大匹配：充要条件为无增广路径（Berge定理）
 - 稳定匹配：每个节点有线性序，稳定匹配算法

本节提要

- 内容1：树的定义及其性质
- 内容2：根树以及有序根树的遍历

树的定义

- 定义：不包含简单回路的连通无向图称为树。
 - 森林（连通分支为树）
 - 树叶/分支点

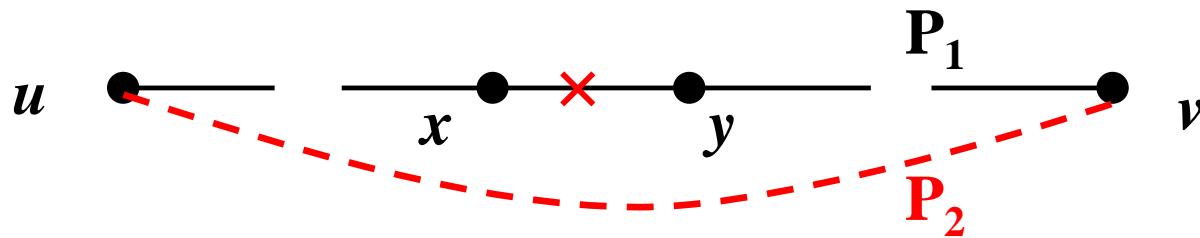


互不同构的6个顶点的树

树中的通路

- 设 T 是树，则 $\forall u, v \in V_T$, T 中存在唯一的 **uv-简单通路**。
- 证明： T 是连通图， $\therefore \forall u, v \in V_T$, T 中存在 **uv-简单通路**。

假设 T 中有两条不同的 **uv-简单通路** P_1, P_2 。不失一般性，存在 $e=(x,y)$ 满足： $e \in P_1$ 但 $e \notin P_2$ ，且在路径 P_1 上 x 比 y 靠近 u 。令 $T^*=T-\{e\}$ ，则 T^* 中包含 P_2 ，于是(P_1 中的 xu -段) + P_2 + (P_1 中的 vy -段) 是 T^* 中的 xy -通路， $\therefore T^*$ 中含 xy -简单通路(记为 P')，则 $P'+e$ 是 T 中的简单回路，与树的定义矛盾。



有关树的几个等价命题

- 设 T 是简单无向图，下列四个命题等价：
 - (1) T 是不包含简单回路的连通图。//树的定义
 - (2) T 中任意两点之间有唯一简单通路。
 - (3) T 连通，但删除任意一条边则不再连通。
 - (4) T 不包含简单回路，但在任意不相邻的顶点对之间加一条边则产生唯一的简单回路。
- 备注：
 - 树是边最少的连通图
 - 树是边最多的无简单回路的图

树中边和点的数量关系

- 设 T 是树，令 $n=|V_T|$, $m=|E_T|$, 则 $\textcolor{red}{m=n-1}$ 。
- 证明. 对 n 进行归纳证明。当 $n=1$, T 是平凡树，结论显然成立。

假设当 $n \leq k$ 时结论成立。

若 $n=k+1$ 。因为 T 中每条边都是割边，任取 $e \in E_T$, $T-\{e\}$ 含 两个连通分支，设其为 T_1, T_2 , 并设它们边数分别是 m_1, m_2 , 顶点数分别是 n_1, n_2 , 根据归纳假设: $m_1=n_1-1, m_2=n_2-1$ 。

注意: $n_1+n_2=n, m_1+m_2=m-1$ 。

$$\therefore m = m_1 + m_2 + 1 = n - 1.$$

连通图边数的下限

□ 顶点数为 n ($n \geq 2$) 的连通图，其边数 $m \geq n-1$ 。

(对于树， $m=n-1$ ，“树是边最少的连通图”)

□ 证明：对 n 进行一般归纳。当 $n=2$ 时结论显然成立。

设 G 是边数为 m 的连通图，且 $|V_G|=n>2$ 。任取 $v \in V_G$ ，令 $G'=G-v$ ，设 G' 有 ω ($\omega \geq 1$) 个连通分支 $G_1, G_2, \dots, G_\omega$ ，且 G_i 的边数和顶点数分别是 m_i 和 n_i 。

我们有 $n = n_1 + n_2 + \dots + n_\omega + 1$, $m \geq m_1 + m_2 + \dots + m_\omega + \omega$ (每个连通分支中至少有一个顶点在 G 中与删除的 v 相邻)。

由归纳假设， $m_i \geq n_i - 1$ ($i=1, 2, \dots, \omega$)。

所以： $m \geq m_1 + m_2 + \dots + m_\omega + \omega \geq n_1 + n_2 + \dots + n_\omega - \omega + \omega = n - 1$ 。

与边点数量关系有关的等价命题

□ 设 T 是简单无向图，下列三个命题等价：

(1) T 是树。

(2) T 不含简单回路，且 $m=n-1$ 。

(3) T 连通，且 $m=n-1$ 。

□ (1) \Rightarrow (2)，已证。

□ (2) \Rightarrow (3)，若不连通，分支数 $\omega \geq 2$ ，各分支为树（无简单回路、连通），则 $m=n-\omega < n-1$ ，矛盾。

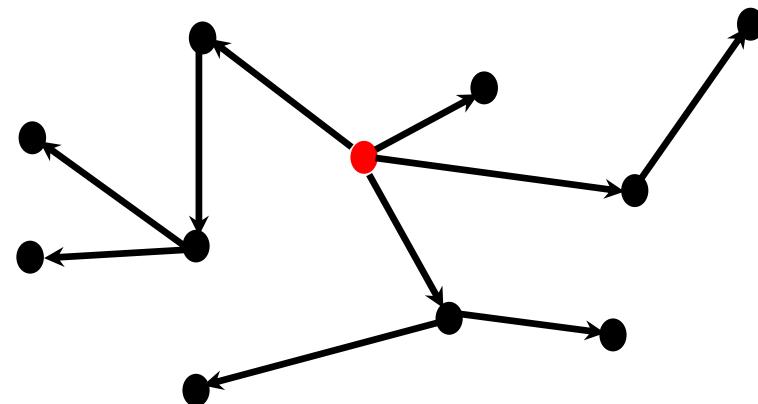
□ (3) \Rightarrow (1)，设 e 是 T 中任意一条边，令 $T'=T-e$ ，且其边数和顶点数分别是 m' 和 n ，则 $m'=m-1=n-2 < n-1$ ， $\therefore T'$ 是非连通图。因此， T 的任意边均不在简单回路中， $\therefore T$ 中无简单回路。

本节提要

- 内容1：树的定义及其性质
 - 树就是不包含简单回路的连通无向图
 - 树是边最少的连通图；也是边最多的无简单回路的图
- 内容2：根树以及有序根树的遍历

根树的定义

- 定义：底图为树的有向图称为**有向树**。
- 定义：若有向树恰含一个入度为**0**的顶点，其它顶点入度均为**1**，则该有向树称为**根树**，那个入度为**0**的顶点称为**根**。
 - 若 v_0 是根树T的根，则对T中任意其它顶点 v_n ，存在唯一的有向 v_0v_n -通路，但不存在 v_nv_0 -通路。



根树的图形表示

- 边上的方向用约定的位置关系表示

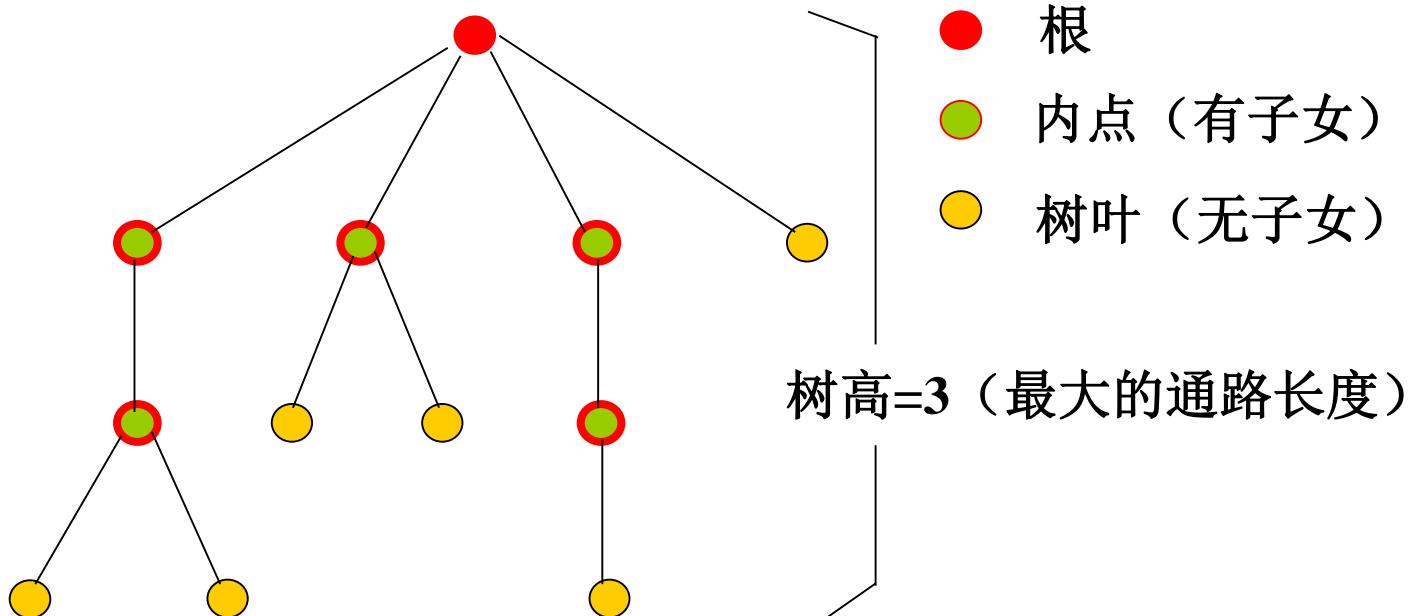
根也是内点，除非它是图中唯一顶点。

第0层

第1层

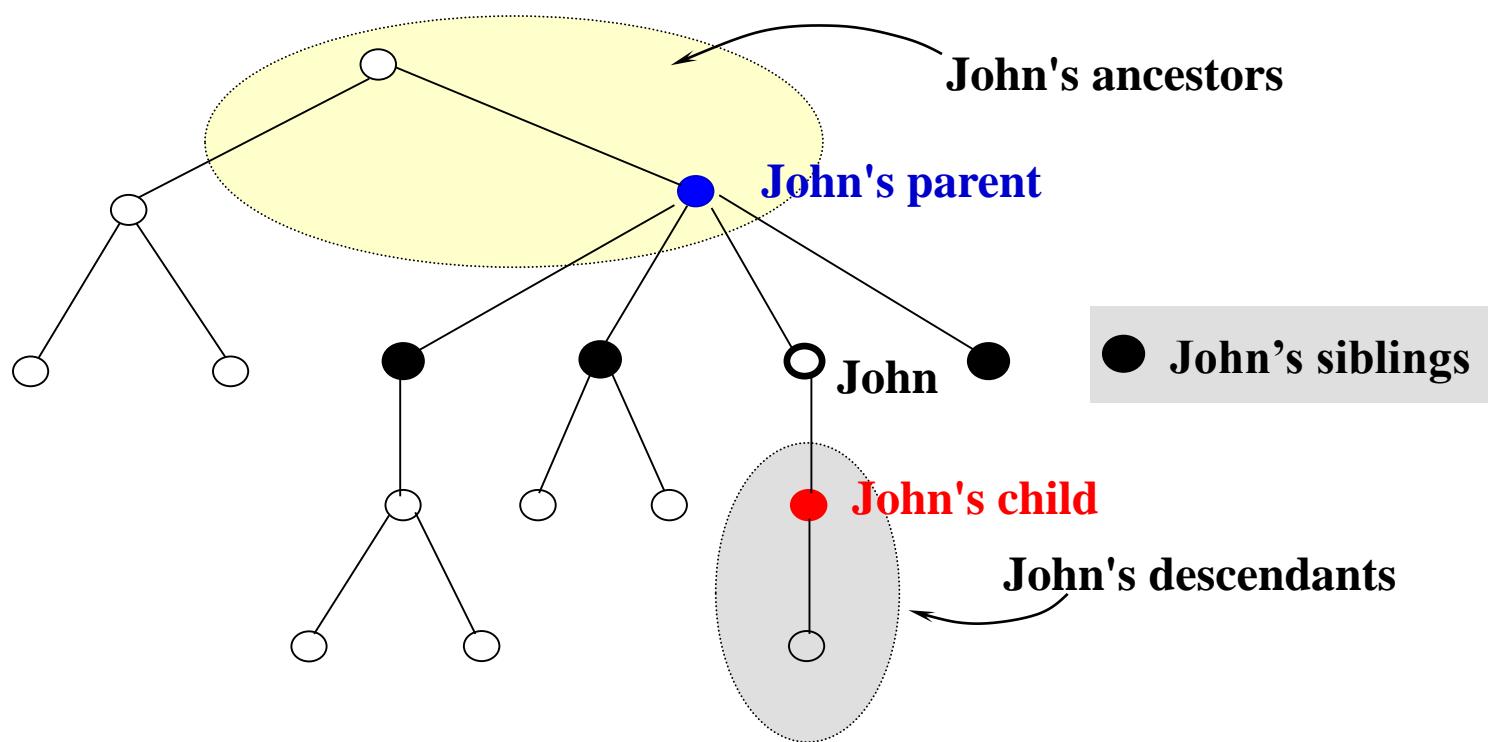
第2层

第3层



根树与家族关系

- 用根树容易描述家族关系，反之，家族关系术语被用于描述根树中顶点之间的关系。

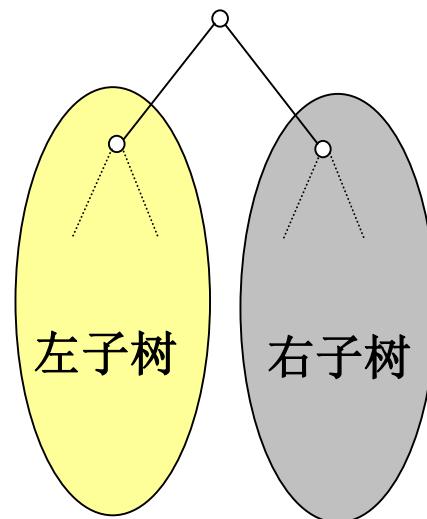


根树的几个术语

- **m元树**: 每个内点至多有m个子女
 - 2元树也称为二叉树
- **满m元树 (*full m-ary tree*)**
 - 每个内点恰好有m个子女
- **完全m元树(*complete m-ary tree*)**
 - 假设有h层: 除第h层外, 其它各层的结点数都达到最大个数, 第h层所有的结点都连续集中在最左边
- **平衡**: 树叶都在h层或(h-1)层, h为树高
- **有序**: 同层中每个顶点排定次序
 - 有序二叉树通常也简称为二叉树

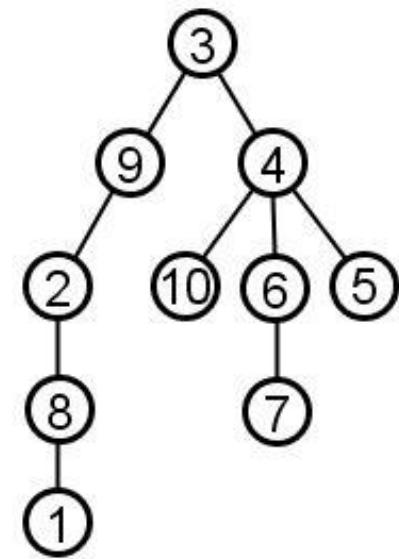
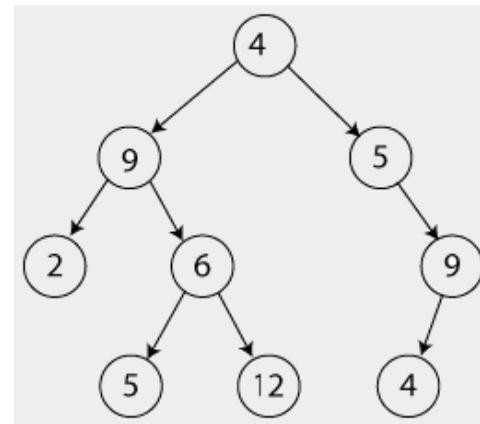
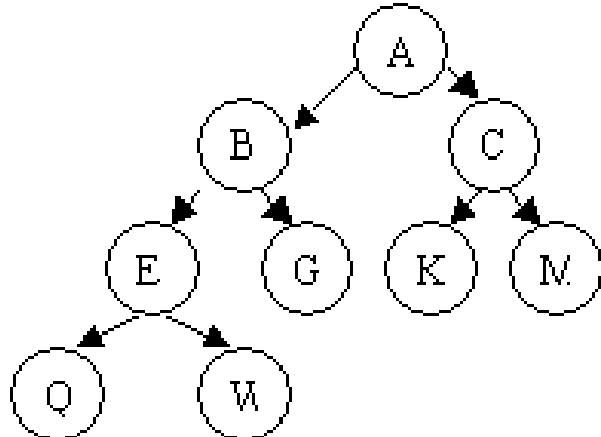
根树的几个术语（续）

- 定义：设 T 是根树， T 中任一顶点 v 及其所有后代的导出子图显然也是根树(以 v 为根)，称为 T 的**根子树**。
- 有序二叉树的子树分为左子树和右子树



例

- 树的高度、各顶点所处的层数
- 满、完全、平衡



满m元树的顶点数

- 设T是满m元树，
 - 若T有n个顶点，则有*i*=(n-1)/m个内点和*l*=[(m-1)n+1]/m个树叶.
 - 若T有*i*个内点，则有n=mi+1顶点和*l*=(m-1)*i*+1个树叶.
 - 若T有*l*个树叶，则有n=(ml-1)/(m-1) 个顶点和*i*=(l-1)/(m-1) 个内点.

$$n-1 = m \times i \quad (\text{入度总数} = \text{出度总数})$$

$$n = i + l \quad (\text{顶点分为内点和树叶})$$

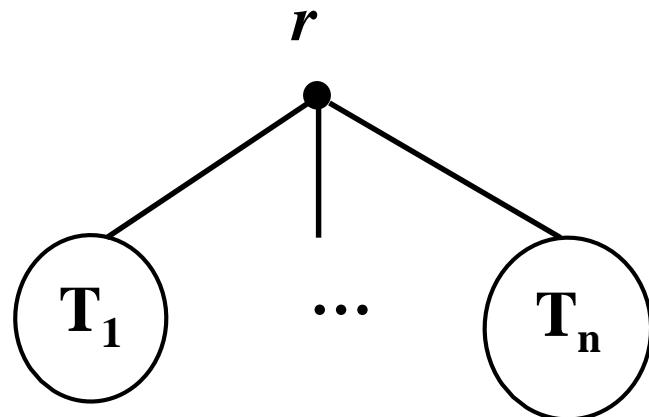
高度为h的m元树的顶点数

- 高度为h的m元树最多有个 m^h 个树叶。
 - 按照高度h进行归纳证明。（第1层顶点最多为m个）
- 若高度为h的m元树有l个树叶，则 $h \geq \lceil \log_m l \rceil$.
 - 如果这棵树是完全的，则有 $h = \lceil \log_m l \rceil$.

有序根树的遍历

□ 前序遍历 (preorder)

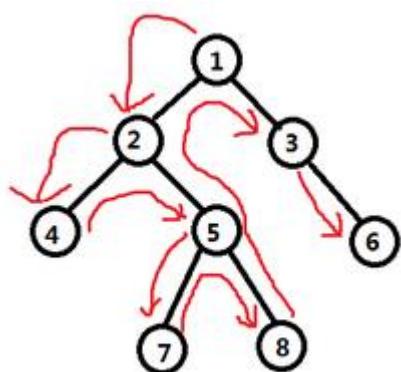
- T 只包含根 r , 则为 r ;
- T 的子树为 T_1, \dots, T_n , 则为 $r, \text{preorder}(T_1), \dots, \text{preorder}(T_n)$



有序根树的遍历

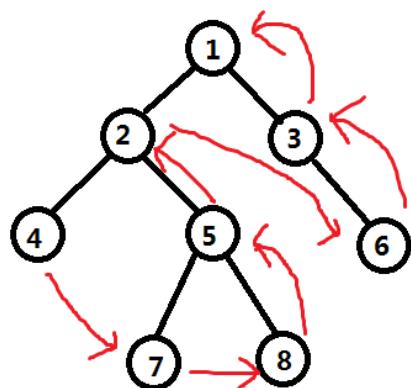
前序遍历

(preorder)



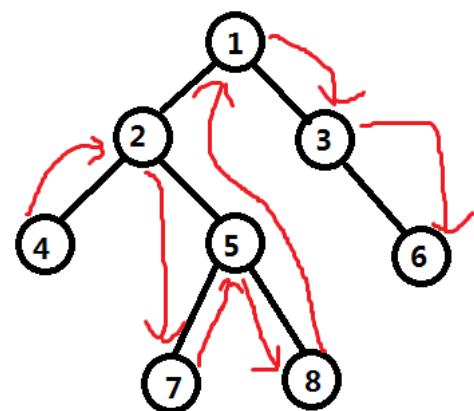
后序遍历

(postorder)



中序遍历

(postorder)



本节小结

- 内容1：树的定义及其性质
 - 树就是不包含简单回路的连通无向图
 - 树是边最少的连通图；也是边最多的无简单回路的图
- 内容2：根树以及有序根树的遍历
 - 根数：一个入度为0的顶点，其它顶点入度均为1
 - 完全树、平衡树、有序树
 - 有序根树的前中后遍历

作业

- 见课程网站

例题

23

- 画出所有**6**个顶点的非同构的无向树
 - 度数序列只能是：
 - 1 1 1 1 1 5; 1 1 1 1 2 4; 1 1 1 1 3 3;
 - 1 1 1 2 2 3; 1 1 2 2 2 2
- 已知一个**7**阶无向树中有三个树叶，一个**3**度顶点（其余三个顶点度数均不为1、3），画出所有满足要求的非同构无向树
 - 度数只能是1112223，考虑度数为3顶点的邻居