

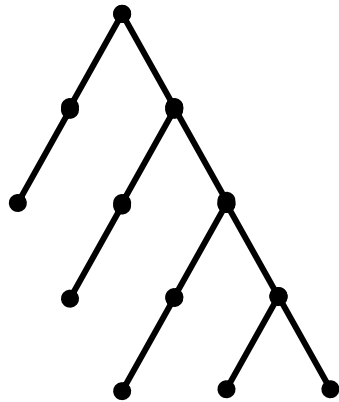
Tree Isomorphism

longhuan@sjtu.edu.cn

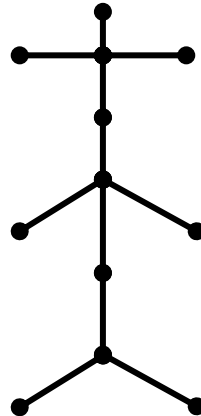
Rooted Tree Isomorphism

树

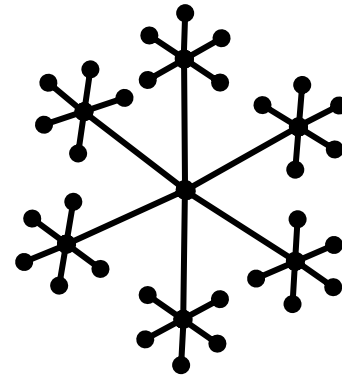
- 树(Tree): 连通无环图。
- 叶子(leaf): 图 G 中度数为1的顶点被称为叶子或终点(end-vertex)。



T_1



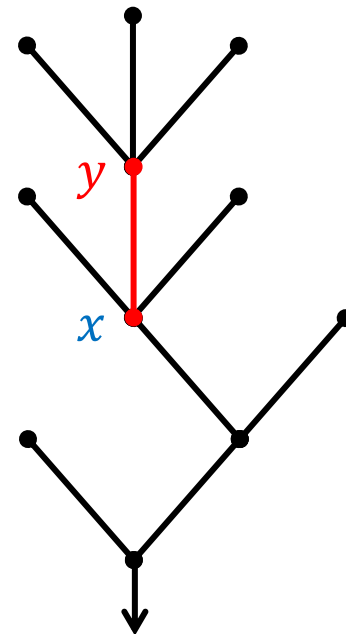
T_2



T_3

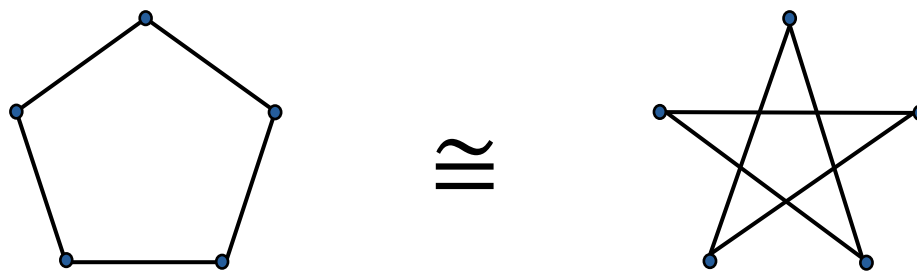
有根树

- **有根树(Rooted tree)**: 二元组 (T, r) 中 T 表示一棵树, $r \in V(T)$ 表示树上的一个特别顶点, 称为根(root)。约定根用箭头标明。
- 对树上的一条边 $\{x, y\} \in E(T)$, 如果 x 是出现在从根 r 到 y 的唯一路径上, 则称 x 是 y 的父亲(father), 相应地称 y 是 x 的儿子(son)。



图同构

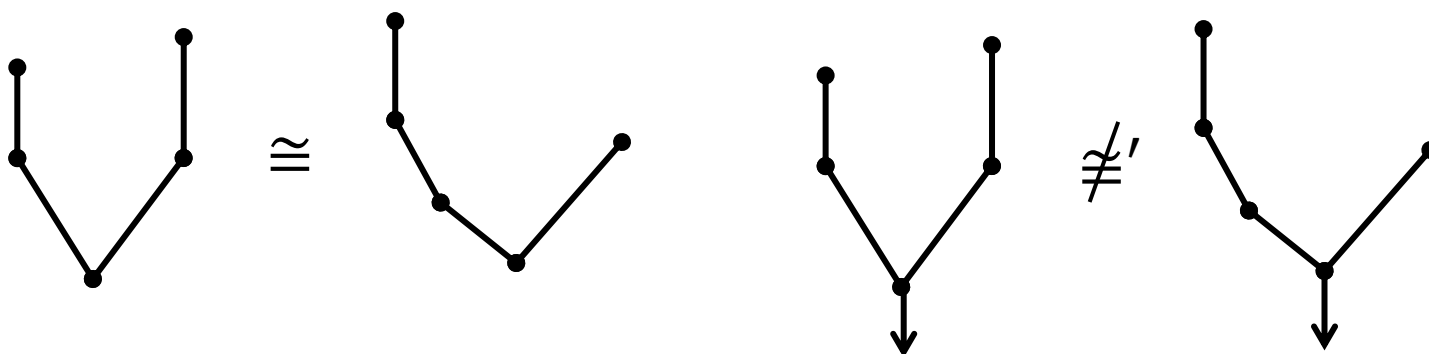
- **图同构(*Graph isomorphism*)**: 若对图 $G = (V, E)$ 以及图 $G' = (V', E')$ 存在双射函数 $f: V \rightarrow V'$, 满足对任意 $x, y \in V$ 都有 $\{x, y\} \in E$ 当且仅当 $\{f(x), f(y)\} \in E'$ 那么我们称图 G 和图 G' 是同构的。



- 一般图之间的同构问题: 尚无有效算法。
- **有根树之间的同构**: 有快速算法。

有根树同构

- **定义:** $(T, r) \cong' (T', r')$:
 - ① $f: V(T) \rightarrow V(T')$ 是 $T \cong T'$,
 - ② $f(r) = r'$ 。
- 例:



\cong' 关系严格地强于 \cong 关系。

有根树同构判定算法

- 思路：将树的比较转化为字符串的比较。
- 字符串比较：字典序(lexicographic order)
对不同序列 $s = s_1s_2 \dots s_n$ 和 $t = t_1t_2 \dots t_m$
 - 如果 s 是 t 的初始序列(即 $t = st_i \dots t_m$)则 $s < t$,
 - 如果 t 是 s 的初始序列(即 $s = ts_j \dots s_m$)则 $t < s$;
 - 否则, 令 i 是 $s_i \neq t_i$ 的最小下标,
 - 若 $s_i < t_i$ 则 $s < t$,
 - 若 $t_i < s_i$ 则 $t < s$ 。
- 例：00<001, 01011<0110。

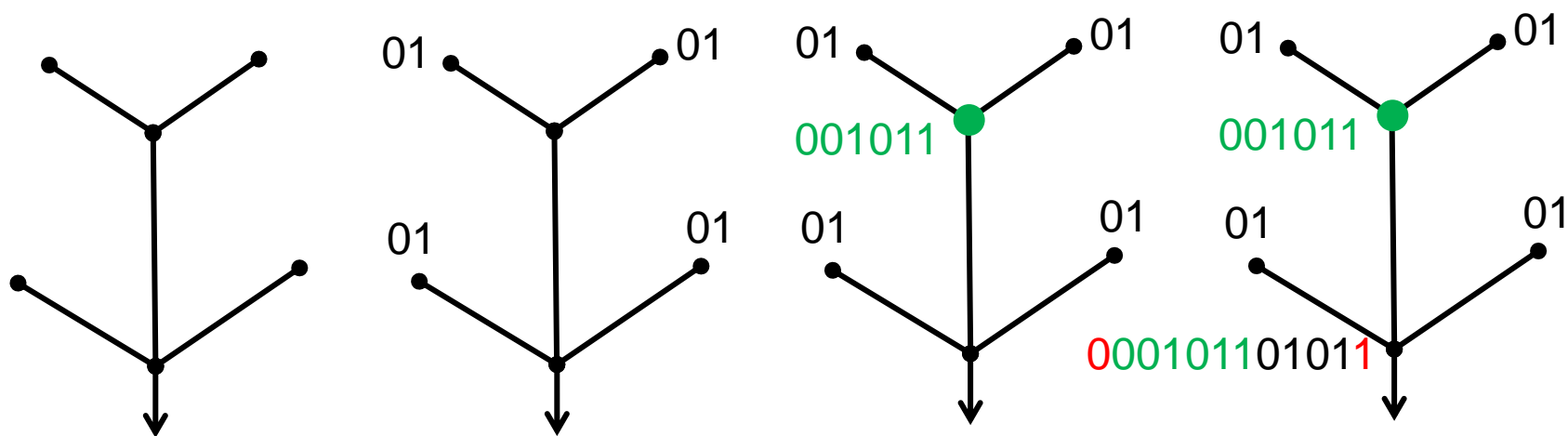
有根树同构判定算法

- 对有根树 (T, r) 如下编码

R1. 所有非根叶结点都赋值为01。

R2. 假设点 v 的儿子节点为 w_1, w_2, \dots, w_k 都已各完成赋值为 $A(w_i)$, 且 $A(w_1) \leq A(w_2) \leq \dots \leq A(w_k)$ 则对 v 节点赋值为 $0A(w_1)A(w_2) \dots A(w_k)1$ 。

根节点 r 的编码就是 (T, r) 的编码, 用 $\#(T, r)$ 表示。



有根树同构判定算法

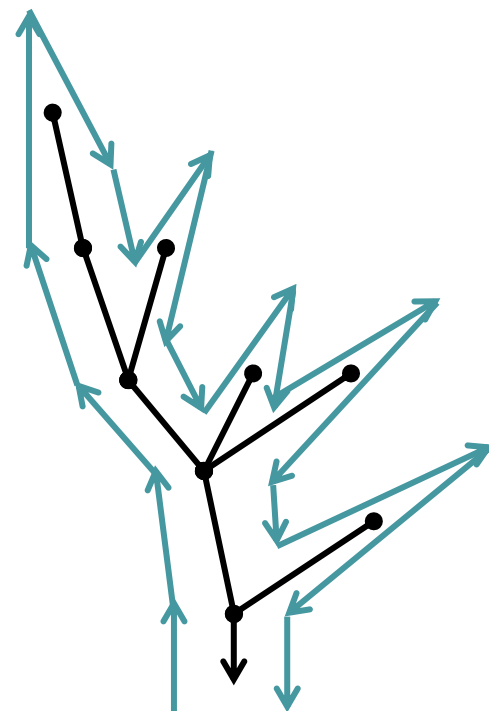
- **性质：** $(T, r) \cong' (T', r')$ 当且仅当它们具有相同的编码。
- **证明：**
 - 充分性：从有根树同构的定义和编码可证。
 - 必要性：**解码**，从编码恢复原始的树结构。
任意有根树的编码必然有0S1的一般形式，其中 $S = S_1 S_2 \dots S_t$ 。
 S_1 是 S 中0,1个数相等的最小前缀。
 S_2 是第二个0,1平衡的最小前缀，等等。
可以据此恢复出有根树，且显然这样的有根树必然是同构的。

从编码恢复原始的树结构

0(0(0(0(01)1(01))1)(01)(01)1)(01)1

0 0 0 0 0 1 1 0 1 1 0 1 1 0 1 1

↑ ↑ ↑ ↑ ↑ ↓ ↓ ↑ ↓ ↓ ↑ ↓ ↑ ↓ ↓ ↑ ↓ ↓

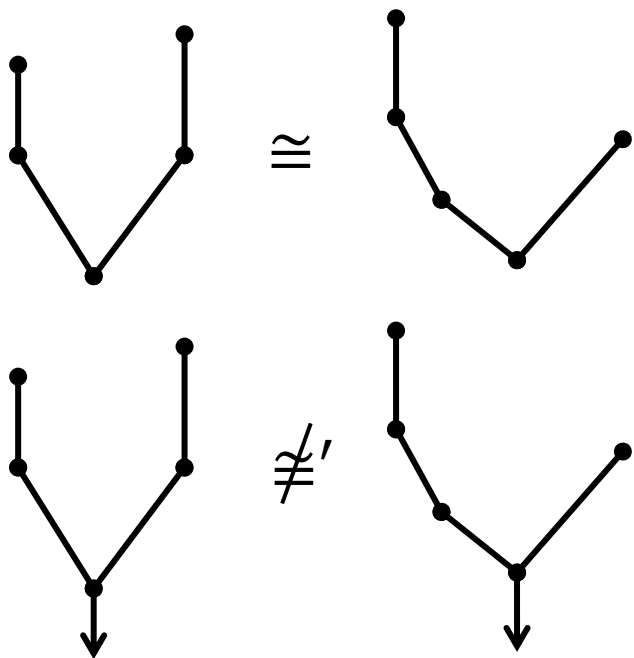


总结

- **性质：** $(T, r) \cong' (T', r')$ 当且仅当它们具有相同的编码。
- 有根树同构存在有效的判定算法。

树同构

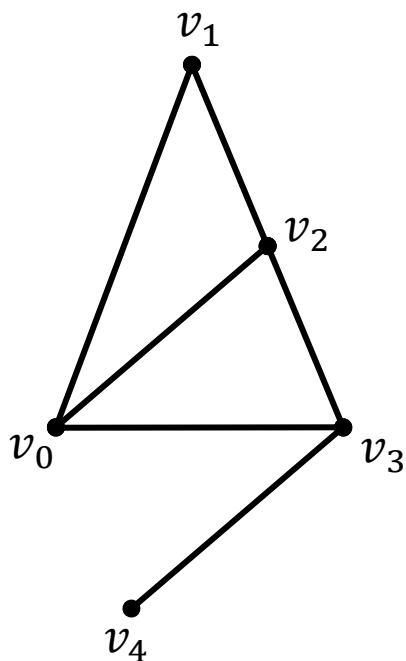
一般树（无根树）同构判定



- 前面确定了有根树的同构判定算法。
- 回顾： \cong' 关系严格地强于 \cong 关系。
- 对一般树(无根树)：找到其中可以用作根的节点，且该根节点在任何同构函数下都被保持。

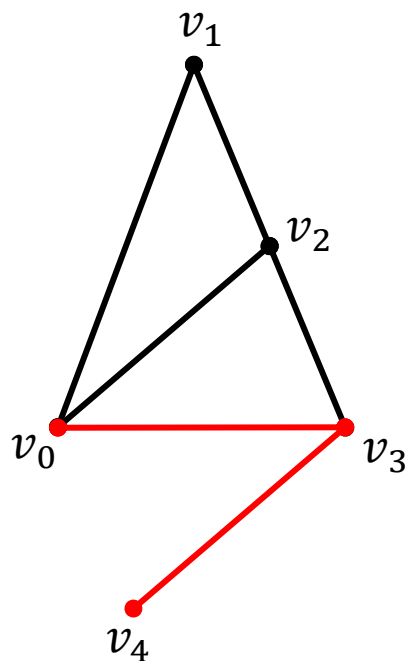
问题规约：一般树同构 \sqsubseteq 有根树同构

为一般树找根



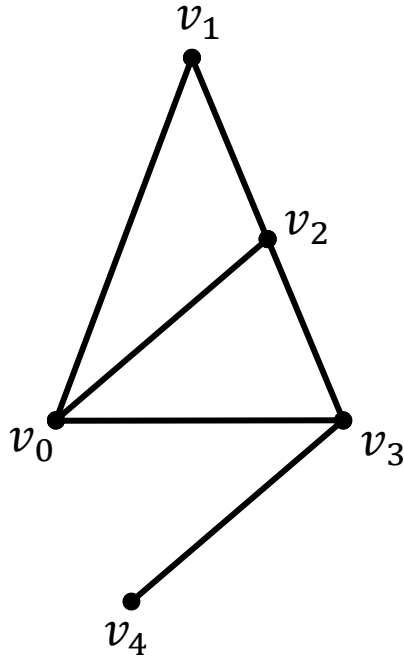
- **距离(Distance)**: 图 G 中的两个顶点 u, v , $dis_G(u, v)$ 表示 u, v 间最短路径的长度。若 u, v 不在一个连通分支里, 定义 $dis_G(u, v) = \infty$ 。
- 例: 左图中 $dis_G(v_0, v_4) = ?$

为一般树找根



- **距离(Distance)**: 图 G 中的两个顶点 u, v , $dis_G(u, v)$ 表示 u, v 间最短路径的长度。若 u, v 不在一个连通分支里, 定义 $dis_G(u, v) = \infty$ 。
- 例: 左图中 $dis_G(v_0, v_4) = 2$

为一般树找根

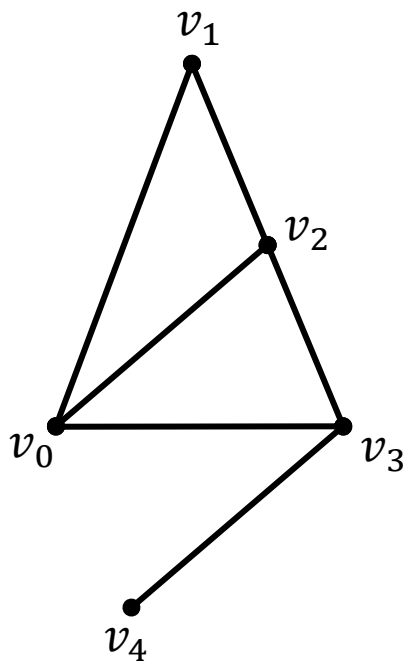


- **偏心率(Excentricity)**: 图 G 及图中的顶点 v , 偏心率定义为:

$$ex_G(v) = \max_{u \in G} dis_G(u, v)$$

- 例: 左图中 $ex_G(v_4) = ?$

为一般树找根



- **偏心率(Excentricity)**: 图 G 及图中的顶点 v , 偏心率定义为:

$$ex_G(v) = \max_{u \in G} dis_G(u, v)$$

- 例: 左图中 $ex_G(v_4) = ?$

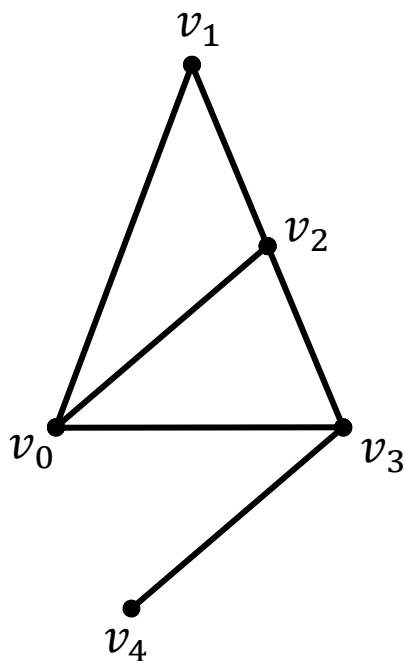
$$dis_G(v_0, v_4) = 2$$

$$dis_G(v_1, v_4) = 3$$

$$dis_G(v_2, v_4) = 2$$

$$dis_G(v_3, v_4) = 1$$

为一般树找根



- **偏心率(Excentricity)**: 图 G 及图中的顶点 v , 偏心率定义为:

$$ex_G(v) = \max_{u \in G} dis_G(u, v)$$

- 例: 左图中 $ex_G(v_4) = 3$

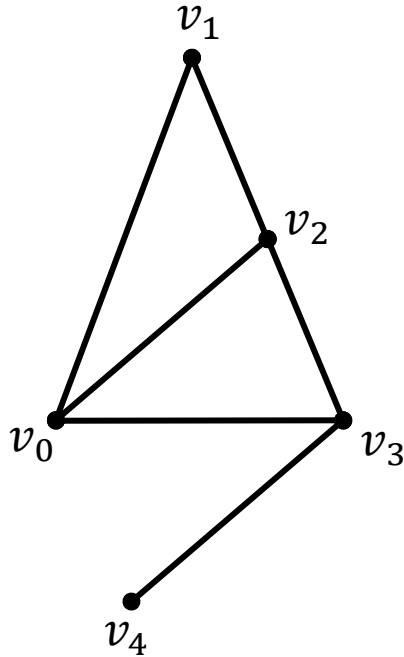
$$dis_G(v_0, v_4) = 2$$

$$dis_G(v_1, v_4) = 3$$

$$dis_G(v_2, v_4) = 2$$

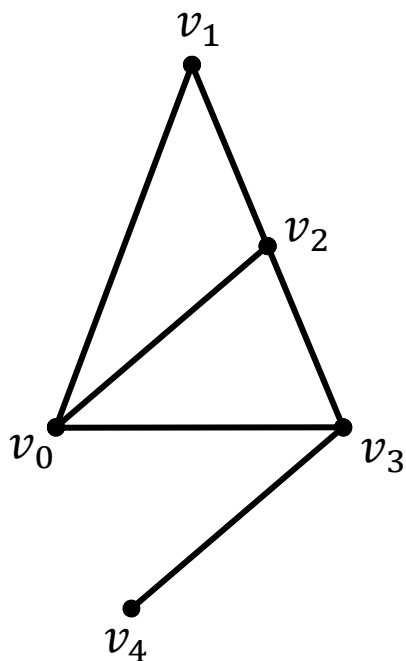
$$dis_G(v_3, v_4) = 1$$

为一般树找根



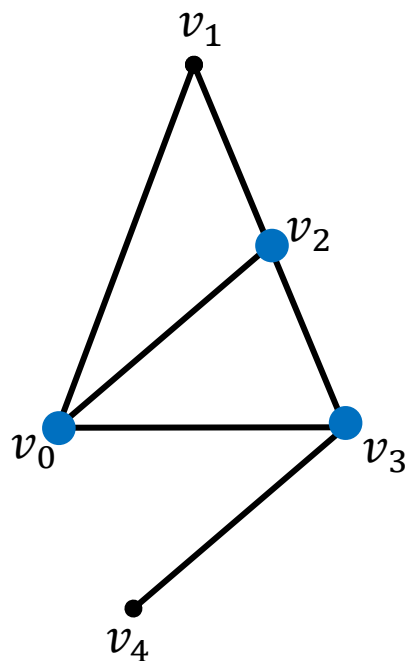
- **中心(Center)**: 图 G 中偏心率最小的顶点集合叫做中心。用符号 $C(G)$ 表示。
- 例: 左图中 $C(G) = ?$

为一般树找根



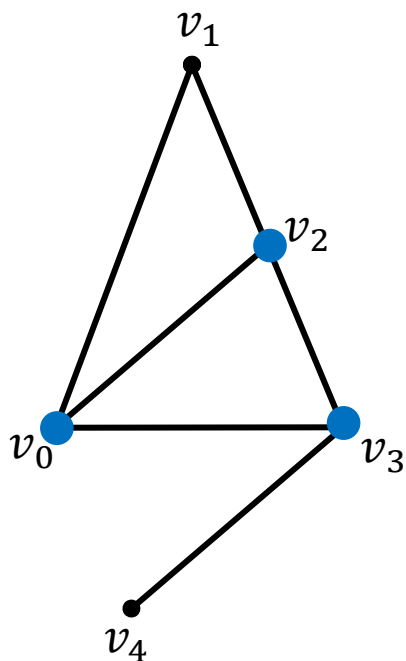
- **中心(Center)**: 图 G 中偏心率最小的顶点集合叫做中心。用符号 $C(G)$ 表示。
- 例: 左图中 $C(G) = ?$
 - $ex_G(v_0) = 2$
 - $ex_G(v_1) = 3$
 - $ex_G(v_2) = 2$
 - $ex_G(v_3) = 2$
 - $ex_G(v_4) = 3$

为一般树找根



- **中心(Center)**: 图 G 中偏心率最小的顶点集合叫做中心。用符号 $C(G)$ 表示。
- 例: 左图中 $C(G) = \{v_0, v_2, v_3\}$
 - $ex_G(v_0) = 2$
 - $ex_G(v_1) = 3$
 - $ex_G(v_2) = 2$
 - $ex_G(v_3) = 2$
 - $ex_G(v_4) = 3$

为一般树找根



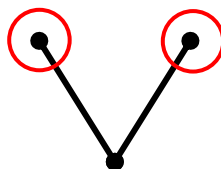
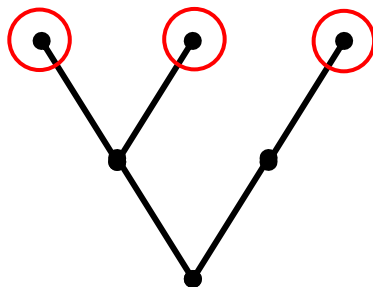
- **中心(Center)**: 图 G 中偏心率最小的**顶点集合**叫做中心。用符号 $C(G)$ 表示。
- 应用: 城市规划。
- 中心可能任意大:
 - 环 C_n , 有 $|C(C_n)| = n$
 - 完全图 K_n , 有 $|C(K_n)| = n$

为一般树找根

- **性质**：对树 $T = (V, E)$ ， $C(T)$ 至多含有2个顶点。且若 $C(T) = \{x, y\}$ ，则 $\{x, y\} \in E$ 。
- 证明：若 $|T| \leq 2$ ，结论显然。否则：
利用树的特殊性：与树上任一点 v 距离最远的点必然是叶子结点。
 - 从 T 构造 T' ： T' 是从 T 中删去所有叶子结点。显然对 T' 上的点 v 有 $ex_T(v) = ex_{T'}(v) + 1$ ，进而 $C(T') = C(T)$ 。
 - 反复以上过程。直至最后剩下一个顶点（ $C(T)$ 是一个顶点）或一条边（ $C(T)$ 是两个顶点）。

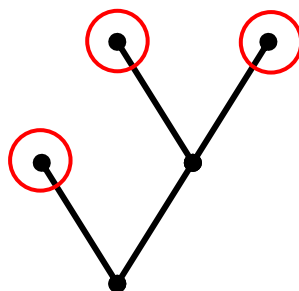
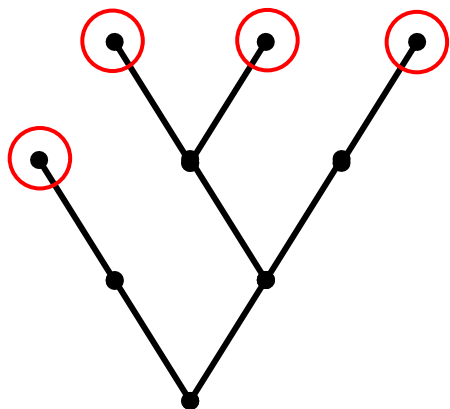
为一般树找根

- 例1:

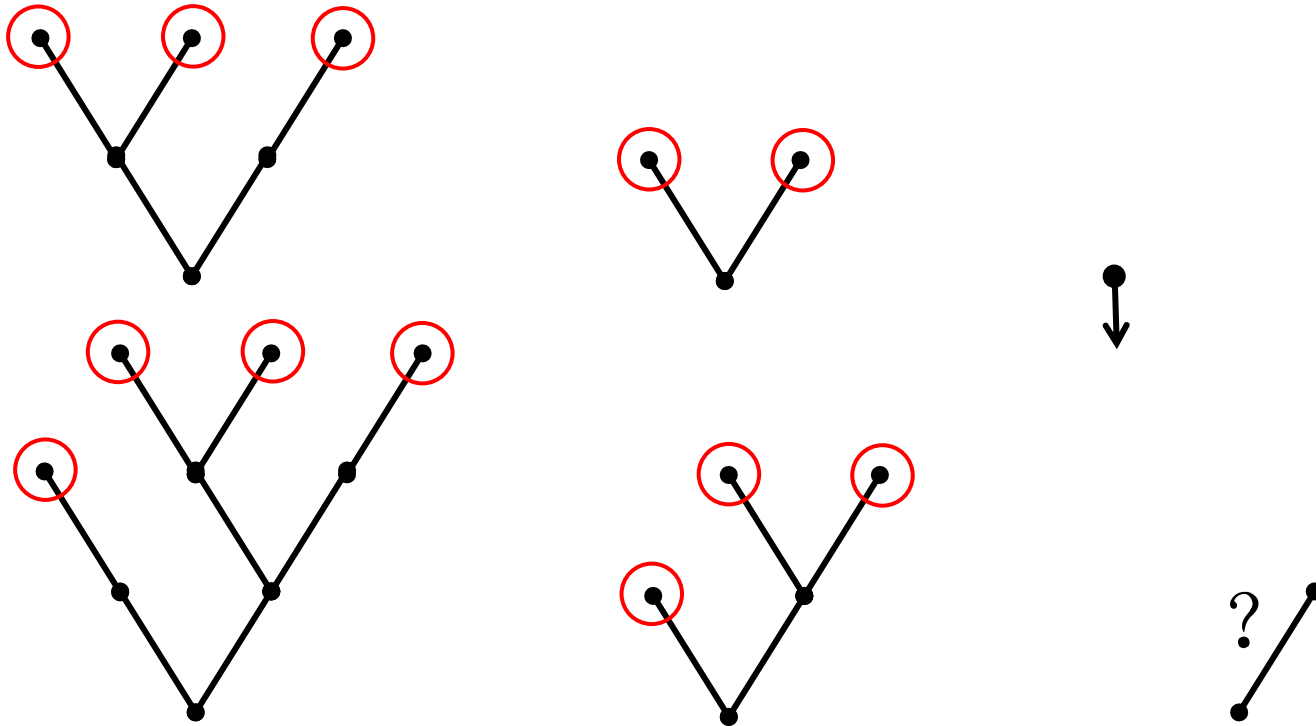


为一般树找根

- 例2:



为一般树找根



用树的中心来完成树到有根树的转化：

- $|C(T)| = 1$ 则中心就是根，
- $|C(T)| = 2$ 的情形如何处理？

树的编码

- $C(T)$ 中只含唯一顶点 v ：输出有根树 (T, v) 的编码 $\#(T, v)$ 。

- $C(T) = \{x_1, x_2\}$ ： $e = \{x_1, x_2\}$

$T - e$ ：必含有正好两个连通分支 T_1, T_2 。不失一般性设 $x_1 \in V(T_1)$, $x_2 \in V(T_2)$ 。

– 计算 $\#(T_1, x_1)$ 和 $\#(T_2, x_2)$ ；

- 如果 $\#(T_1, x_1) \leq \#(T_2, x_2)$ ，输出 $\#(T, x_1)$ ；
- 否则，输出 $\#(T, x_2)$ 。

$\#T$ = 上述过程的输出

- 验证: $T \cong T'$ 当且仅当 $\#T \cong \#T'$ 。
- 证明: (与有根树证明相似)

树同构问题

- 树同构存在有效判定算法
 - 找中心
 - 定根
 - 有根树编码
 - 编码比较