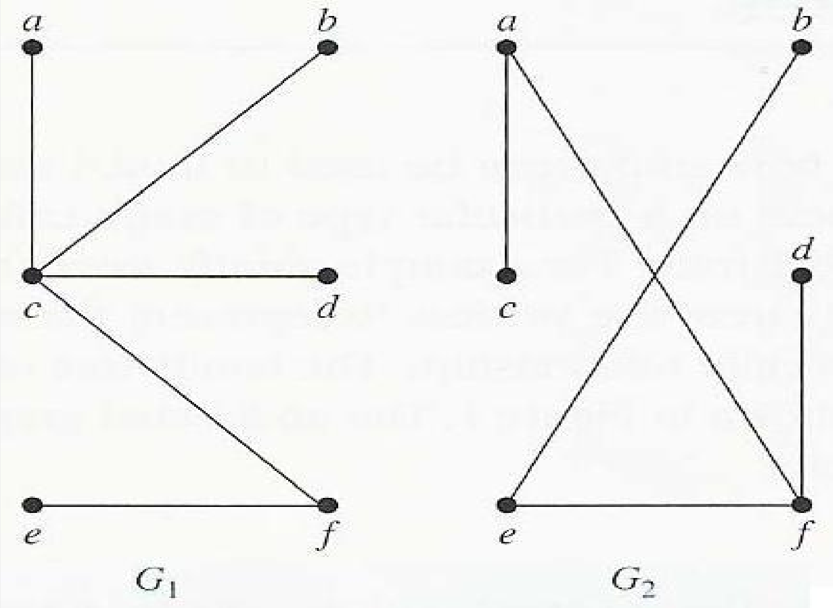
树

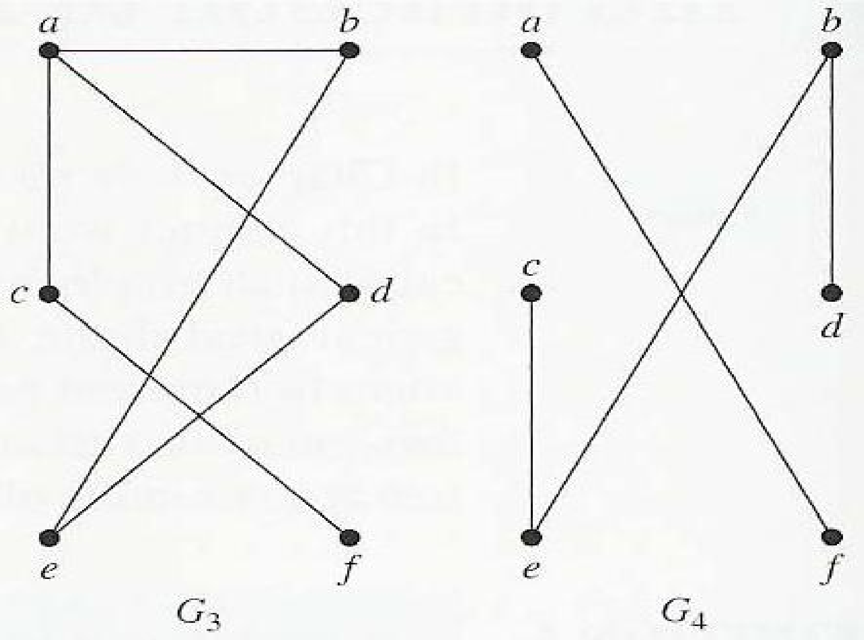
- 定义1　连通无回路的图称为树，树中度为1的点称为树叶，度大于1的点称为分枝点或内点，每个连通分支均为树的图称为森林。

Example: Trees



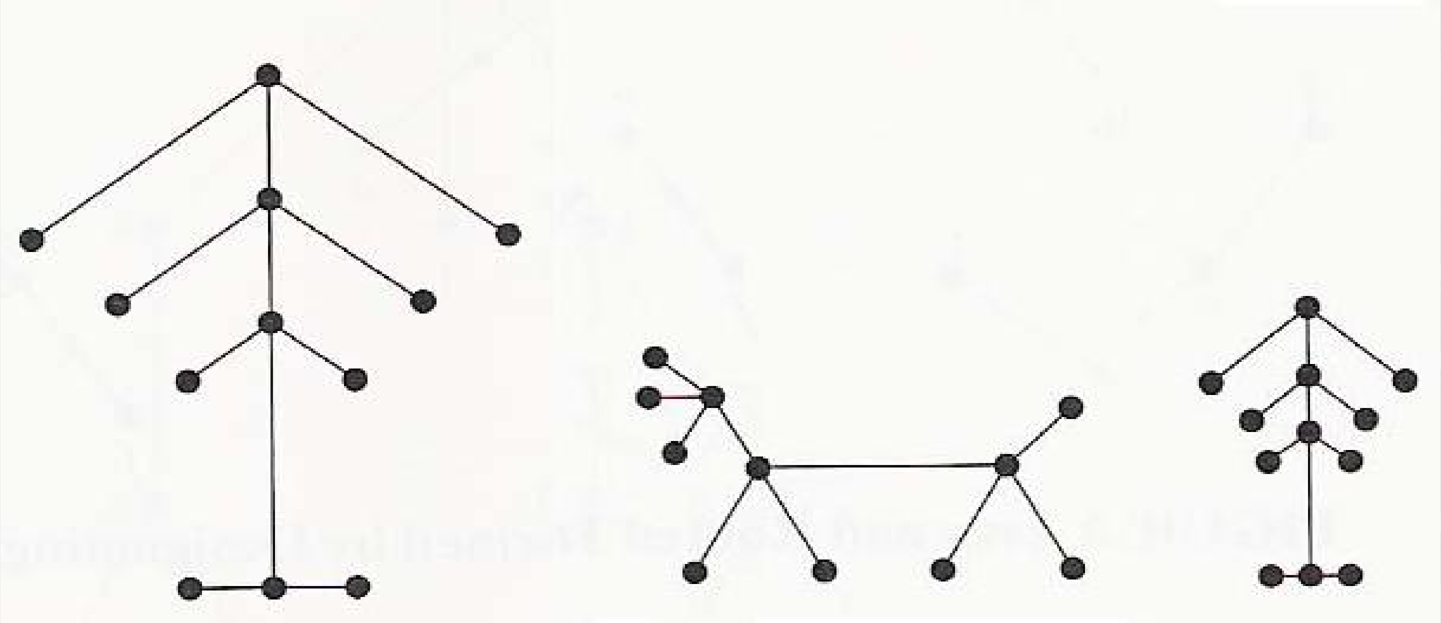
- G_1 and G_2 are trees. Both are connected graph with no simple circuits.

Example: Not Trees



- G_3 is not a tree..
- G_4 is not a tree.

Example: Forest



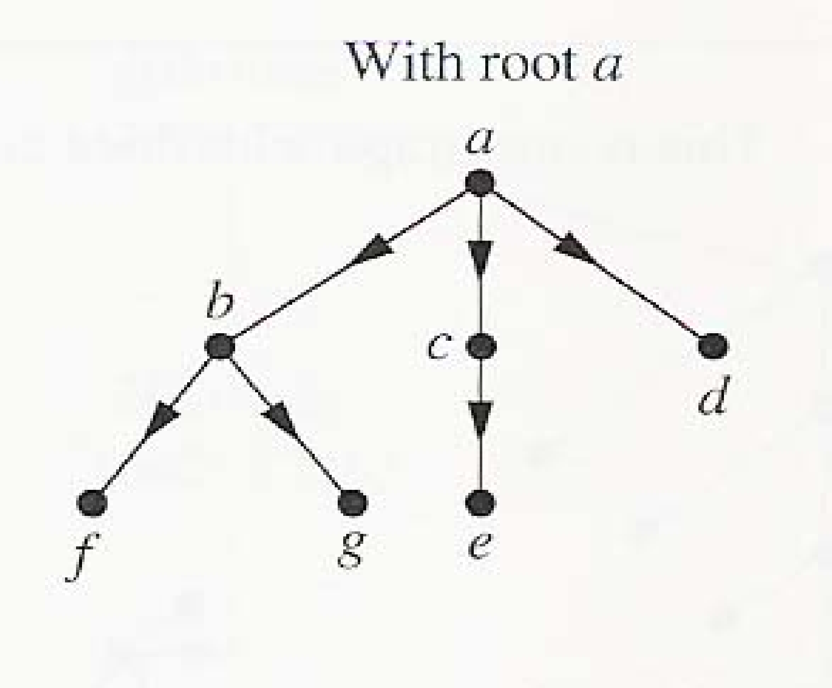
One graph with three connected components

- **Theorem**: Let G be a graph with n nodes and e edges
    - 1.G is a tree (connected, acyclic)
    - 2. G is acyclic and $e = n - 1$
    - 3.G is connected and $e = n - 1$
    - 4.G is acyclic and if any two non-adjacent points are joined by an edge, the resulting graph has exactly one cycle
    - 5. G is connected, but if any edge is deleted, it will be non-connected
    - 6. Every two nodes of G are joined by a unique path
- **Theorem**: there exists at least two nodes of degree one for every tree.

有向树

- 定义1　设D是一个有向图，如果在不考虑弧的方向时D是一棵树(即D的底图是一棵树)则称D为一棵有向树。
- 定义 2　若一棵有向树中恰有一个顶点的入度为 0 ，其余所有顶点的入度均为1，则称该有向树为有根树(或树形图)，入度为 0 的顶点称为根。

A Rooted Tree



Rooted Trees

- We can change an unrooted tree into a rooted tree by choosing any vertex as the root.
- Different choices of the root produce different trees.

- 有根树的画法
- 定义 3　儿子，父亲，兄弟，子孙，祖先；从根到某一顶点的路长称为该顶点的路长或层数，从根到树叶的最大层数，称为有根树的高。

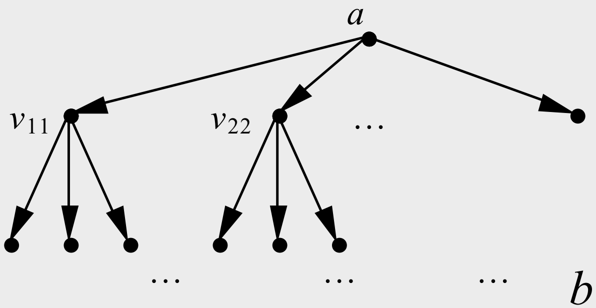
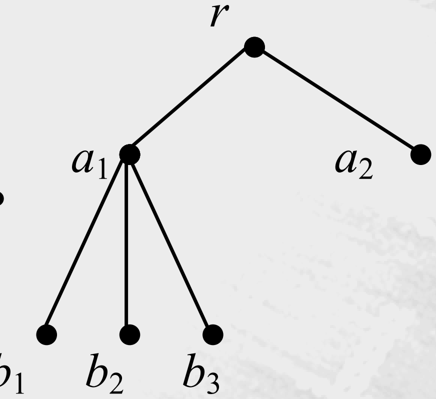
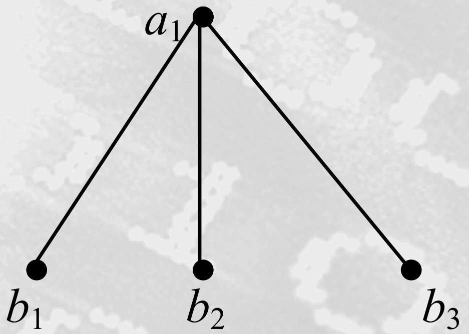


图 3.2

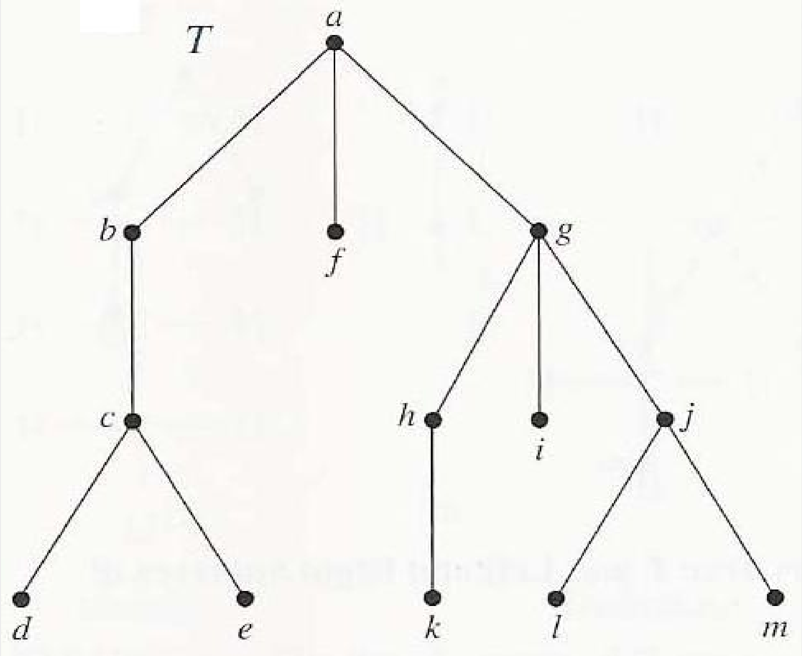


(*a*)



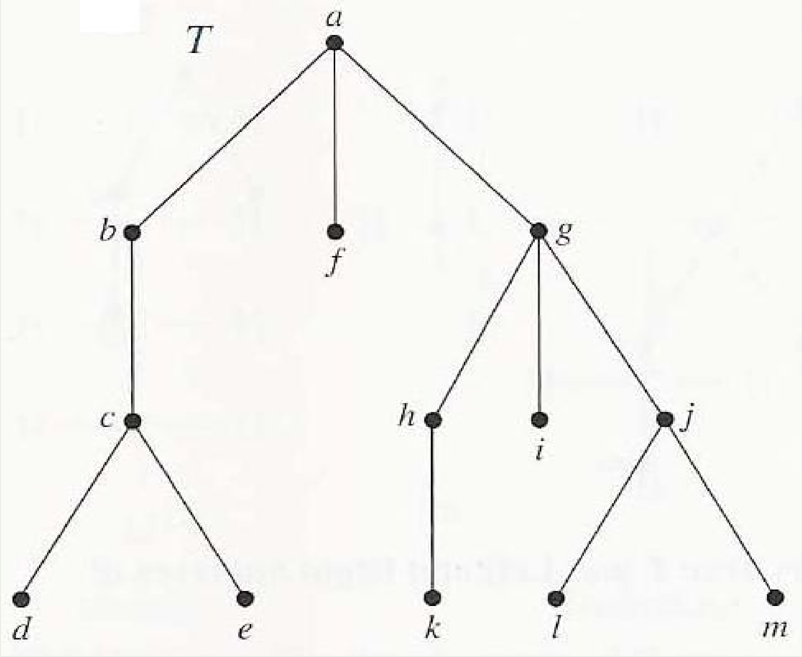
(*b*)

Example: Using Terminology



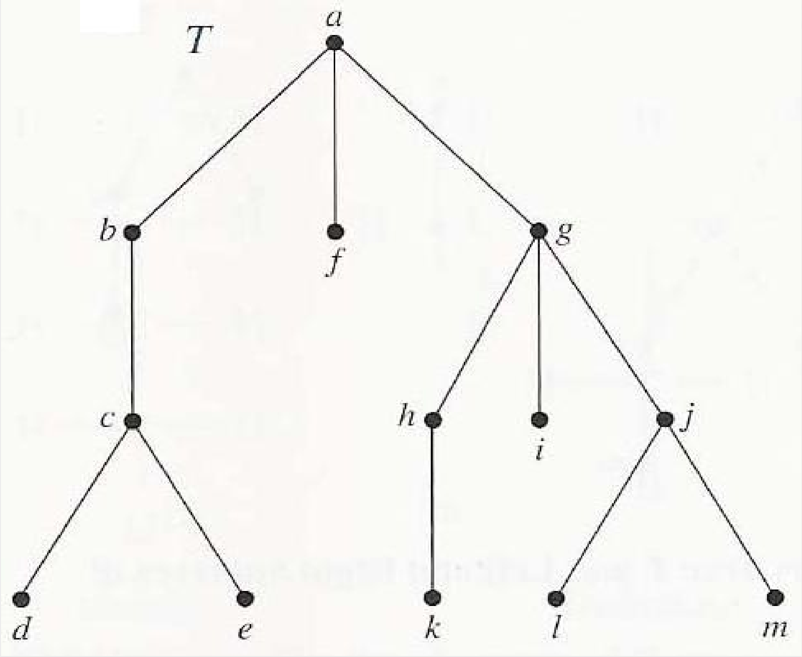
- T is a rooted tree with root a.
- The parent of vertex c is b.
- The children of g are h, i, and j.

Example: Using Terminology



- The siblings of h are i and j.
- The ancestors of e are c, b, and a.
- The descendant of b are c, d, and e.

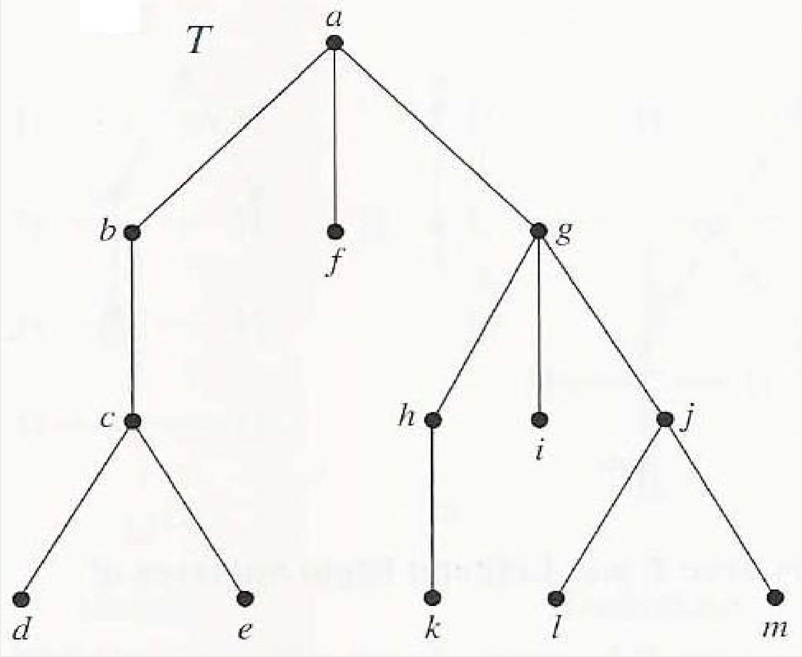
Example: Using Terminology



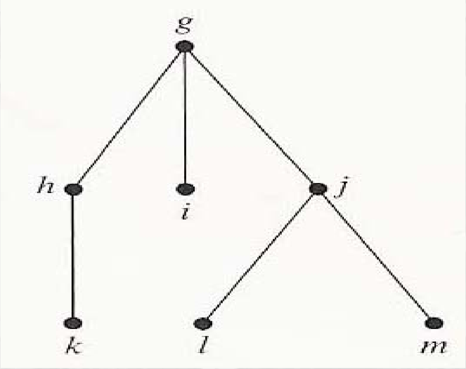
- The internal vertices are a, b, c, g, h, and j.
- The leaves are d, e, f, i, k, l, and m.

- 定义 4　设u是有根树T的一个顶点，V_u是u及其子孙构成的顶点集，V_u的导出子图称为T的以u为根的子树。

Example: Using Terminology



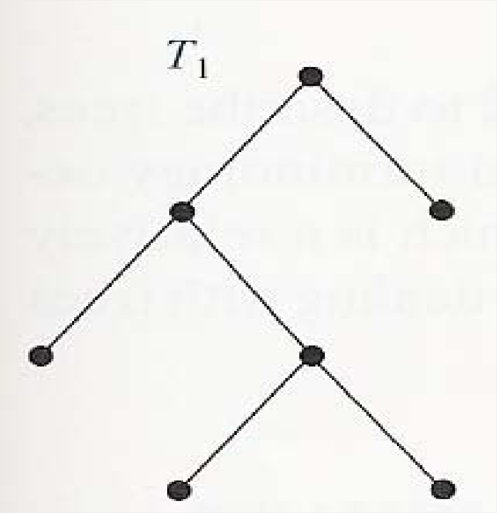
- The subtree rooted at g is



m-ary Tree

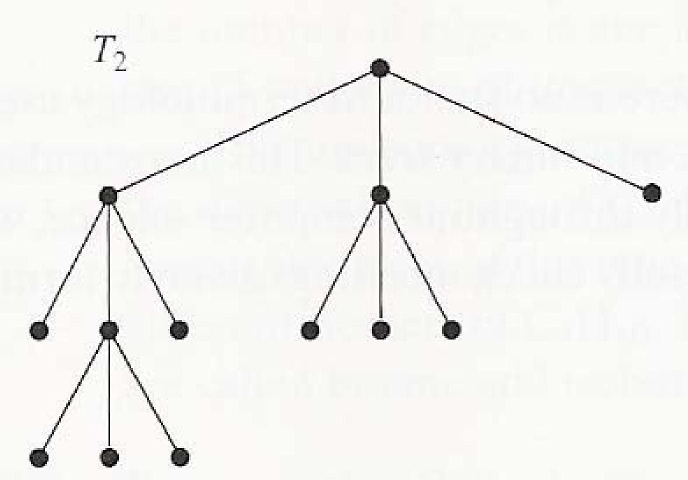
- A rooted tree is called an m-ary tree if every internal vertex has no more than m children. The tree is called a full m-ary tree if every internal vertex has exactly m children. An m-ary tree with m = 2 is called a binary tree.

Example of m-ary tree



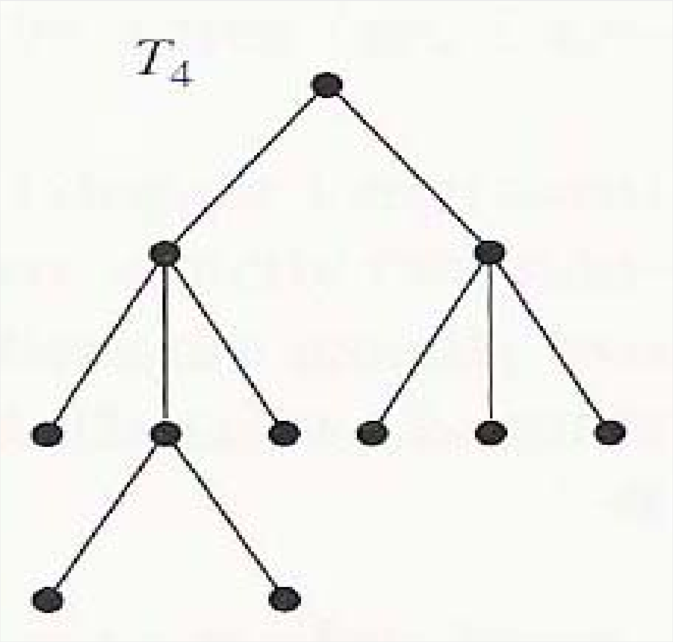
- T_1 is a full binary tree.
- Each of its internal vertices has two children

Example of m-ary tree



- T_2 is a full 3-ary tree.
- Each of its internal vertices has three children

Example of m-ary tree



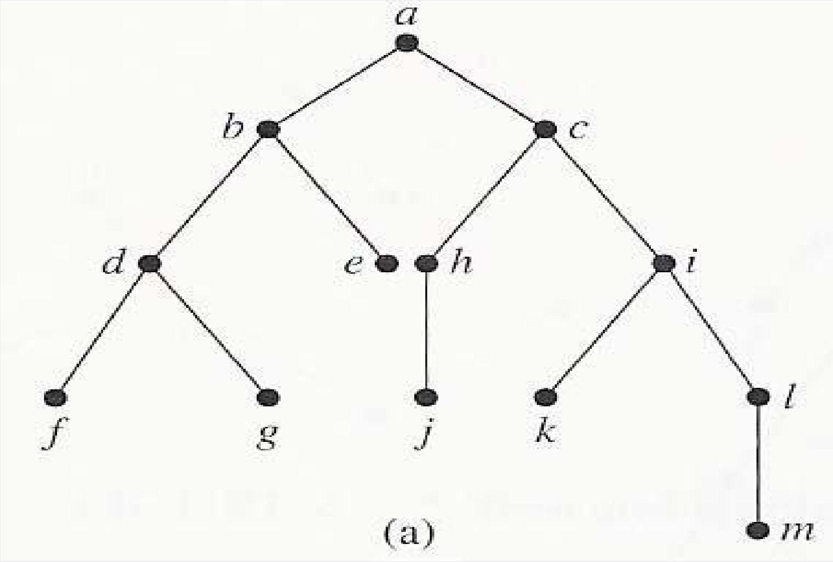
- T_4 is not a full m-ary tree for any m.
- Some of its internal vertices has 2 children and others have 3.

Ordered Rooted Tree

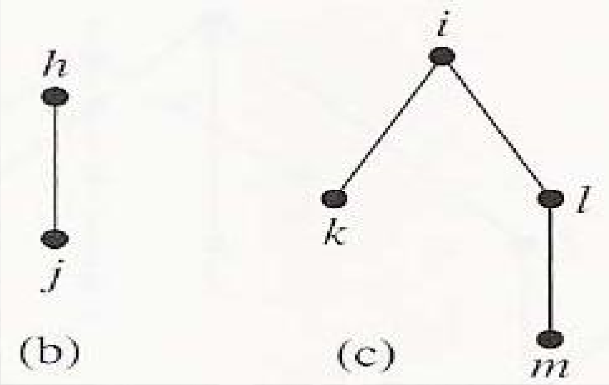
- An ordered rooted tree is a rooted tree where the children of each internal vertex are ordered.
- In an ordered binary tree, if an internal vertex has two children, the first child is called the left child and the second child is called the right child.
- The tree rooted at the left child of a vertex is called the left subtree of this vertex, and the tree rooted at the right child of a vertex is called the right subtree of the vertex.

Example

- The left child of d is f and the right child is g.
- The left and right subtrees of c are

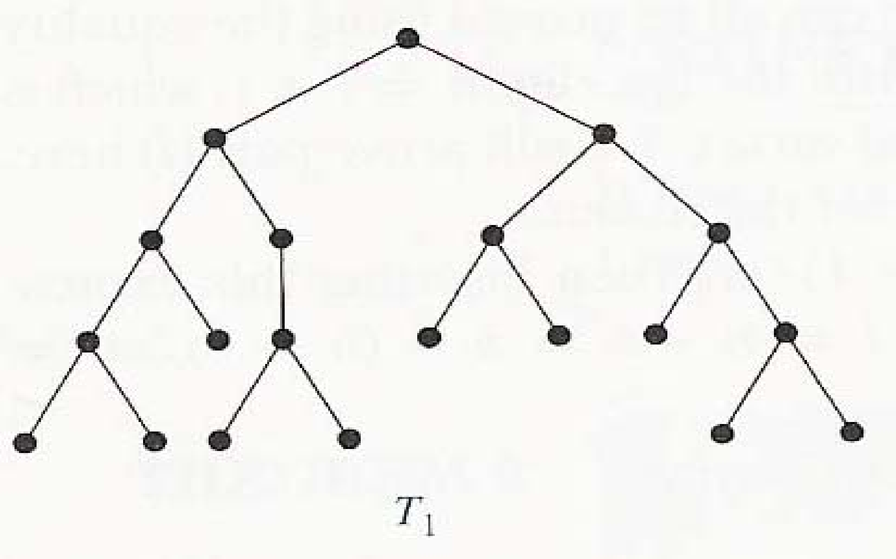


(a)



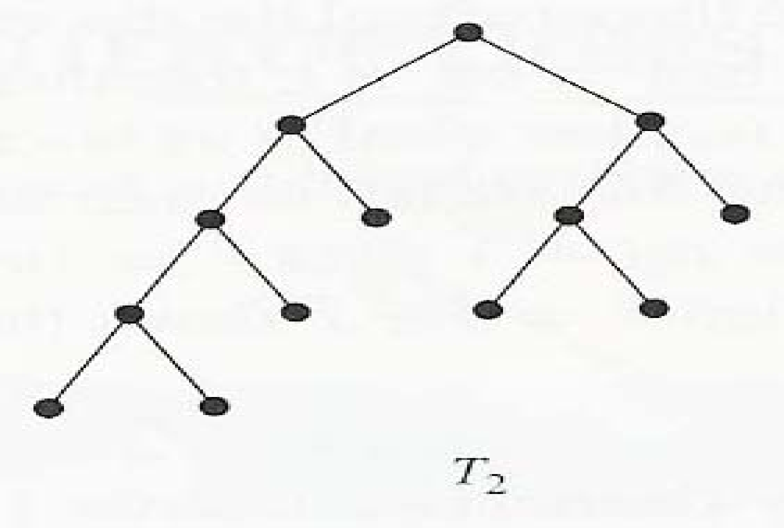
(b) (c)

- A rooted m-ary tree of height is **balanced** if all leaves are at levels h or h − 1.



T_1

- T_1 is **balanced.**
- All its leaves are at levels 3 and 4.



T_2

- T_2 is not **balanced**.
- It has leaves at levels 2, 3, and 4.

生成树

- 定义1　若图*G*的生成子图*T*是树，则称*T*为*G*的生成树。

- 定理1　*G*是连通图当且仅当*G*有生成树。

- 权图*G*中带权最小的生成树称为最小生成树

- Kruskal算法
- 输入：简单连通图G，权函数*w* 。
- 输出：最小生成树T

Kruskal算法

- (1)选取G的一边e_1，使$w(e_1)=\min\{w(e)|e\in E\}$，令$E_1=\{e_1\}$
- (2)若已选出$E_i=\{e_1,\ \ldots,\ e_i\}$，那么，从$E-Ei$中选取一边$e_{i+1}$，使

  (I)$E_i\cup\{e_{i+1}\}$的导出子图中不含回路；

  (II)$w(e_{i+1})=\min\{w(e)|e\in E-E_i, E_i\cup\{e\}$的导出子图无回路$\}$
- (3)若e_{i+1}存在，令$E_{i+1}=E_i\cup\{e_{i+1}\}$，$i+1\to i$，转(2)，若$e_{i+1}$不存在，则停止。