Discrete Mathematics (ITPC-309)

Trees : Part I

Mrs. Sanga G. Chaki

Department of Information Technology

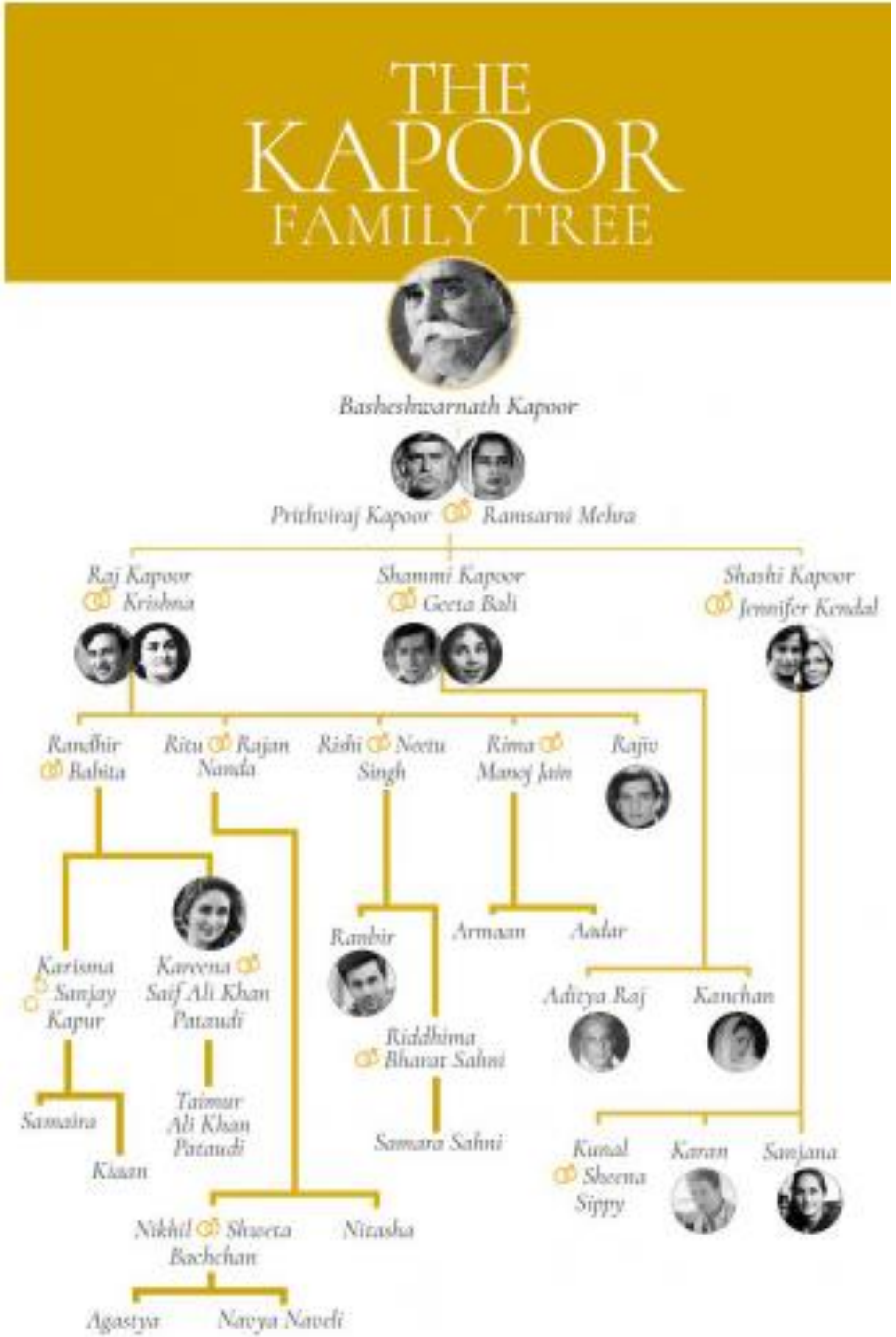
Dr. B. R. Ambedkar National Institute of Technology, Jalandhar

Contents

- Tree: Definition,
- Rooted tree,
- Properties of trees,

Trees

1. A particular type of graph
2. So named because such graphs resemble trees (inverted trees).
3. A lot of the terminology is also related to trees.
4. Example: family trees are graphs that represent genealogical charts.
5. Family trees use vertices to represent the members of a family and edges to represent parent–child relationships.

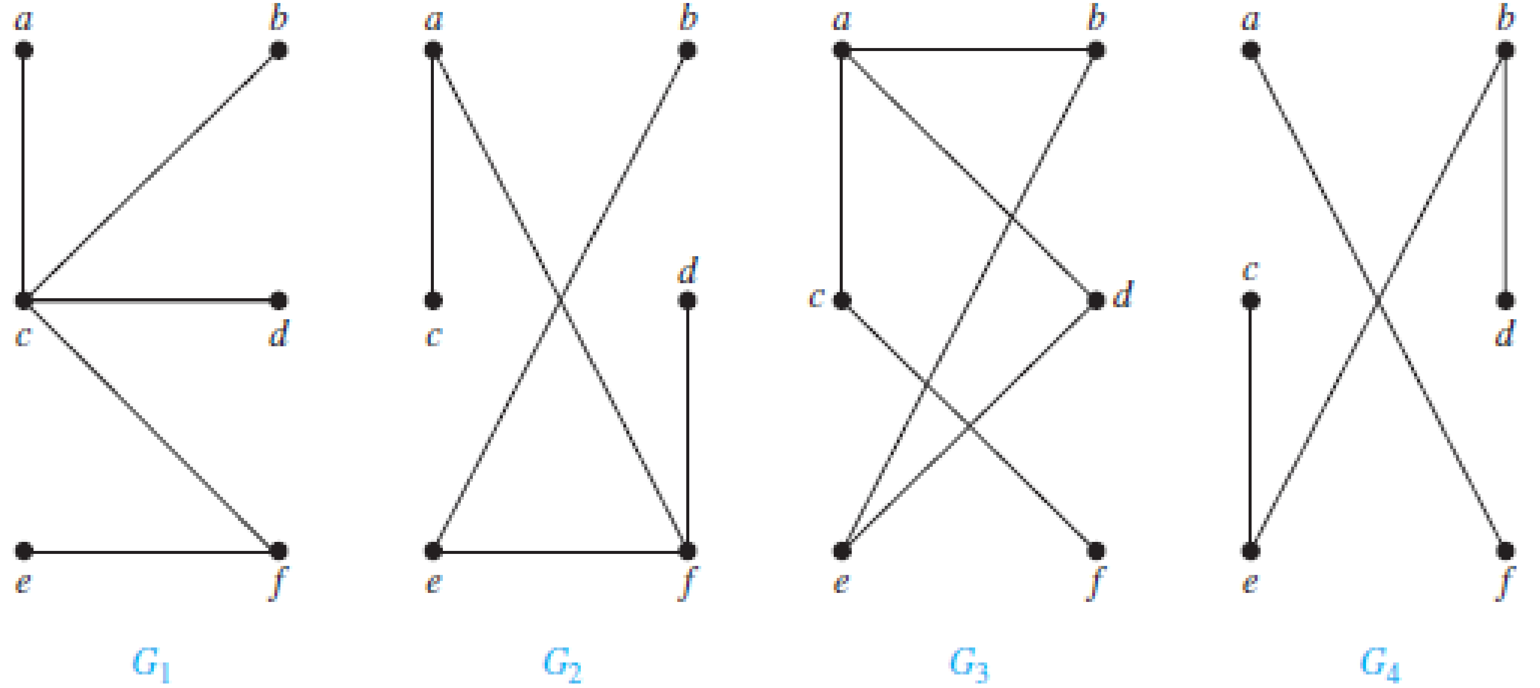


Trees

1. **Definition: A tree is a connected undirected graph with no simple circuits**.
2. A tree cannot contain multiple edges or loops.
3. Any tree must be a simple graph.
4. **An undirected graph is a tree if and only if there is a unique simple path between any two of its vertices**.
   - Proof: Since a tree is a connected graph, there must be a simple path between any two vertices (say x an y)
   - This simple path must also be unique, since if this was not the case, the two paths between x and y would form a circuit.
   - This would contradict the definition of a tree.
   - Hence proved.

Trees

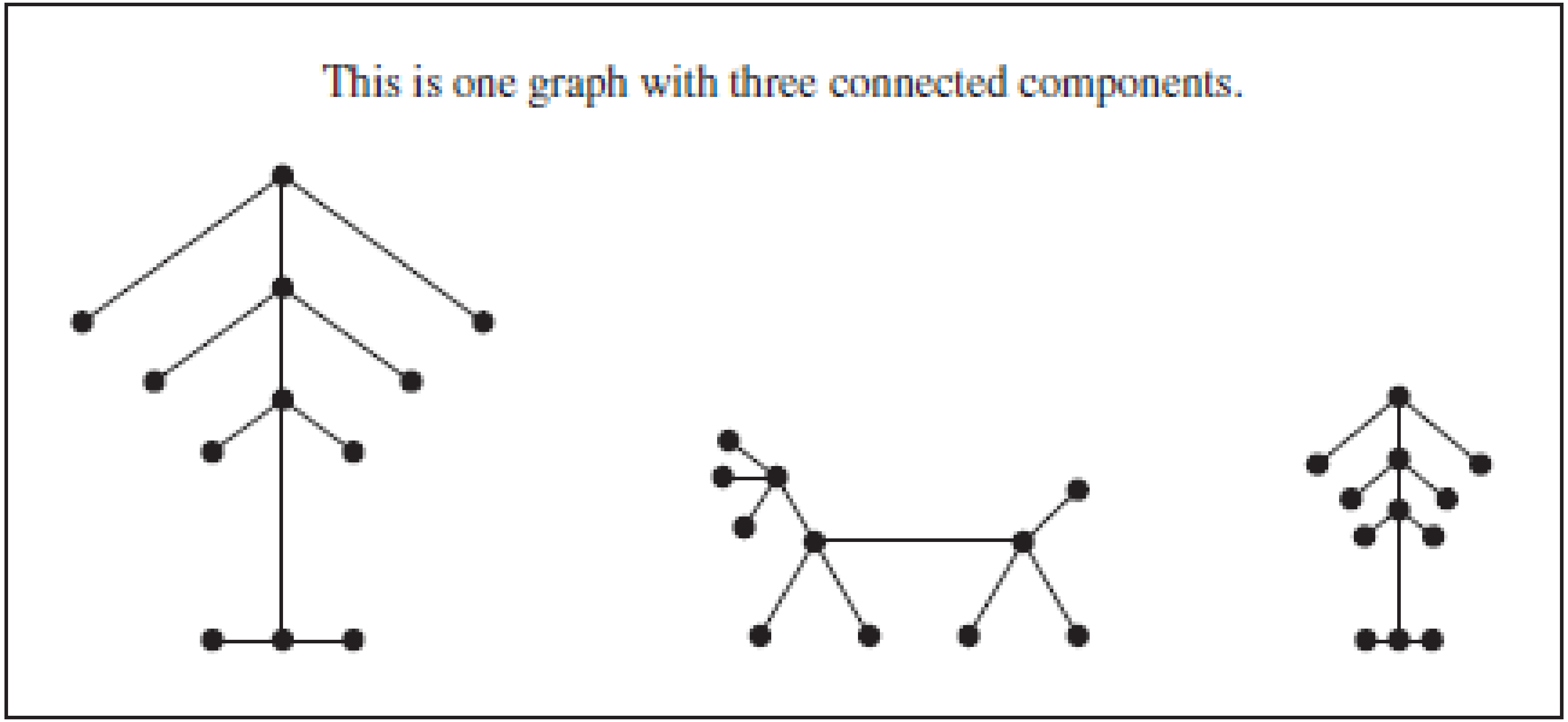
1. Example: Are the following trees? See if it is connected. See if simple circuit present.



2. G1 and G2 are trees, because both are connected graphs with no simple circuits.
3. G3 is not a tree because e, b, a, d, e is a simple circuit in this graph.
4. G4 is not a tree because it is not connected.

Forests

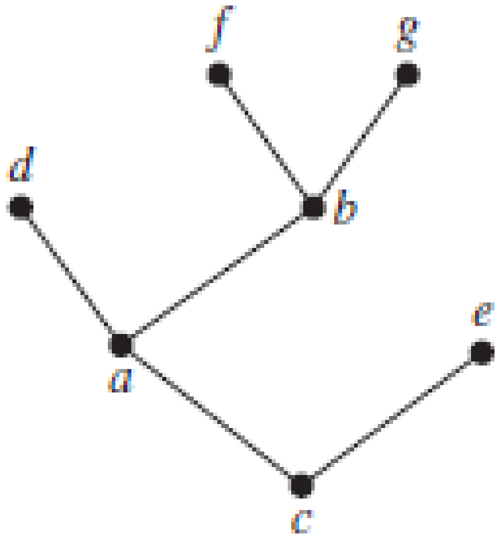
1. Graphs containing no simple circuits that are not connected are called forests
2. Each of their connected components is a tree



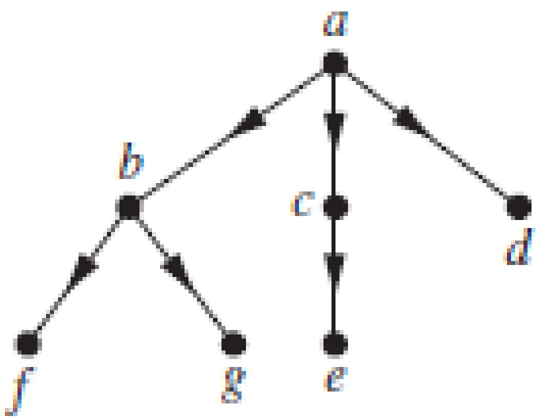
Rooted Trees

1. **A rooted tree is a tree in which one vertex has been designated as the root and every edge is directed away from the root.**
2. We can change an unrooted tree into a rooted tree by choosing any vertex as the root.
3. Different choices of the root produce different rooted trees.
4. We usually draw a rooted tree with its root at the top of the graph.
5. The arrows indicating the directions of the edges in a rooted tree can be omitted, because the choice of root determines the directions of the edges.

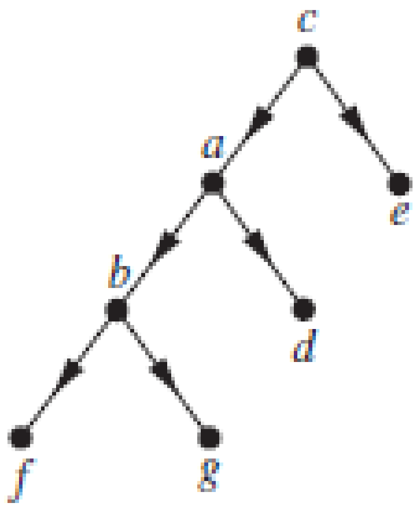
T



With root *a*

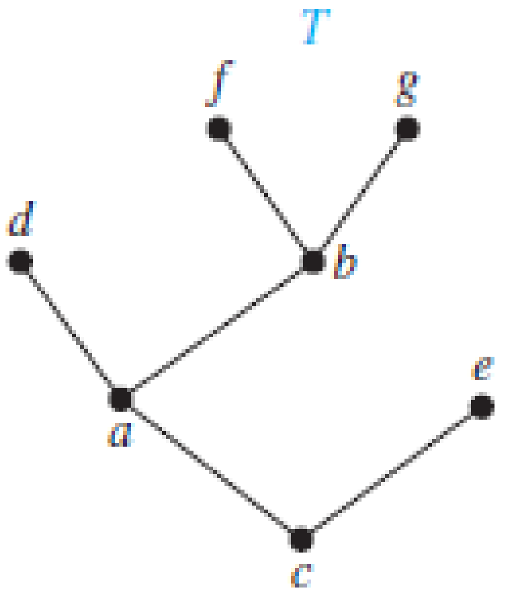


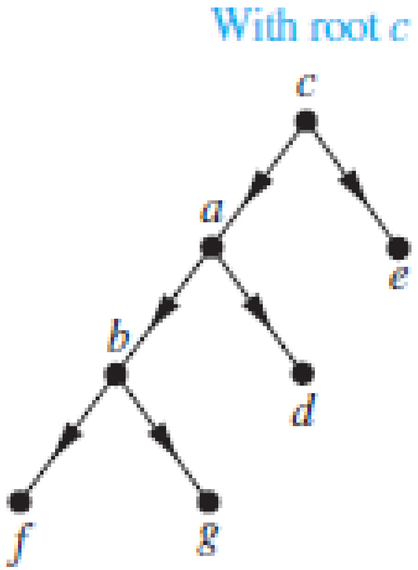
With root *c*

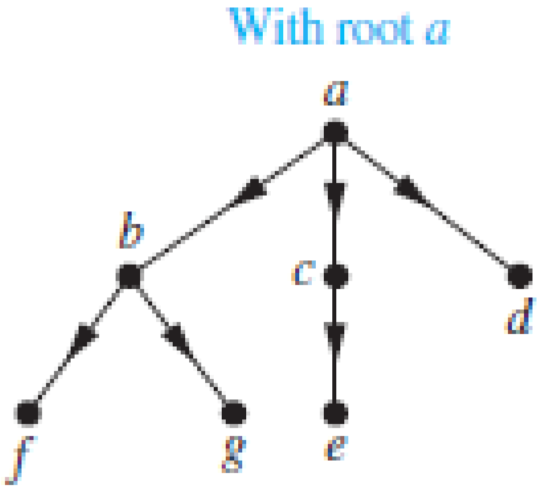


Rooted Trees

1. Example: Change the given unrooted tree T in rooted tree, with roots at each of the 5 other vertices, except for a and c.







Rooted Tree Terminology

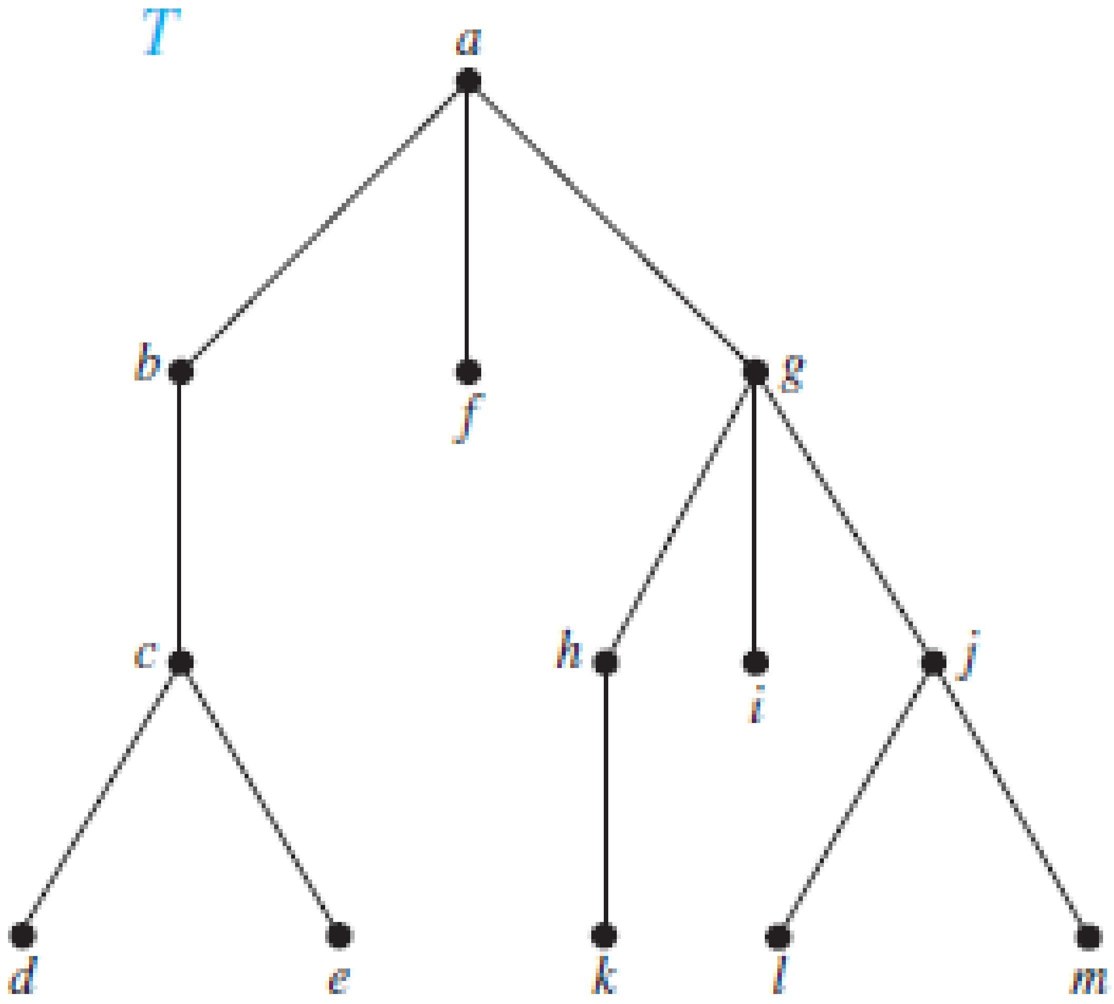
1. Suppose that T is a rooted tree.
2. If v is a vertex in T other than the root, the **parent** of v is the unique vertex u such that there is a directed edge from u to v.
3. When u is the parent of v, v is called a **child** of u.
4. Vertices with the same parent are called **siblings**.
5. The **ancestors** of a vertex other than the root are the vertices in the path from the root to this vertex, excluding the vertex itself and including the root
   - its parent, its parent's parent, and so on, until the root is reached
6. The **descendants** of a vertex v are those vertices that have v as an ancestor.

Rooted Tree Terminology

1. A vertex of a rooted tree is called a **leaf** if it has no children.
2. Vertices that have children are called **internal vertices**.
3. The root is an internal vertex unless it is the only vertex in the graph, in which case it is a leaf.
4. If ***a*** is a vertex in a tree, the **subtree** with ***a*** as its root is the subgraph of the tree consisting of ***a*** and its descendants and all edges incident to these descendants.

Rooted Tree

1. Example: In the below rooted tree T (with root a), find
   - the parent of c,
   - the children of g,
   - The siblings of h,
   - all ancestors of e,
   - all descendants of b,
   - all internal vertices, and
   - all leaves.
   - What is the subtree rooted at g?

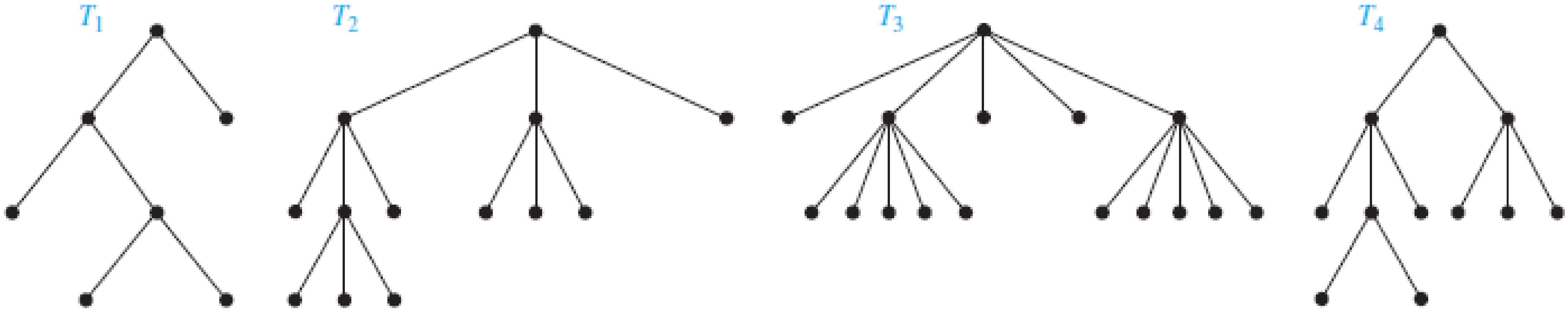


M-ary Rooted Tree

1. A rooted tree is called an **m-ary tree** if every internal vertex has no more than m children.
2. The tree is called a full m-ary tree if every internal vertex has exactly m children.
3. An m-ary tree with m = 2 is called a binary tree.
4. So in a binary tree, each internal vertex can have a maximum of 2 children.
5. Note: m-ary tree vs full m-ary tree

Rooted Tree

1. Example: Are the rooted trees given below full m-ary trees for some positive integer m?

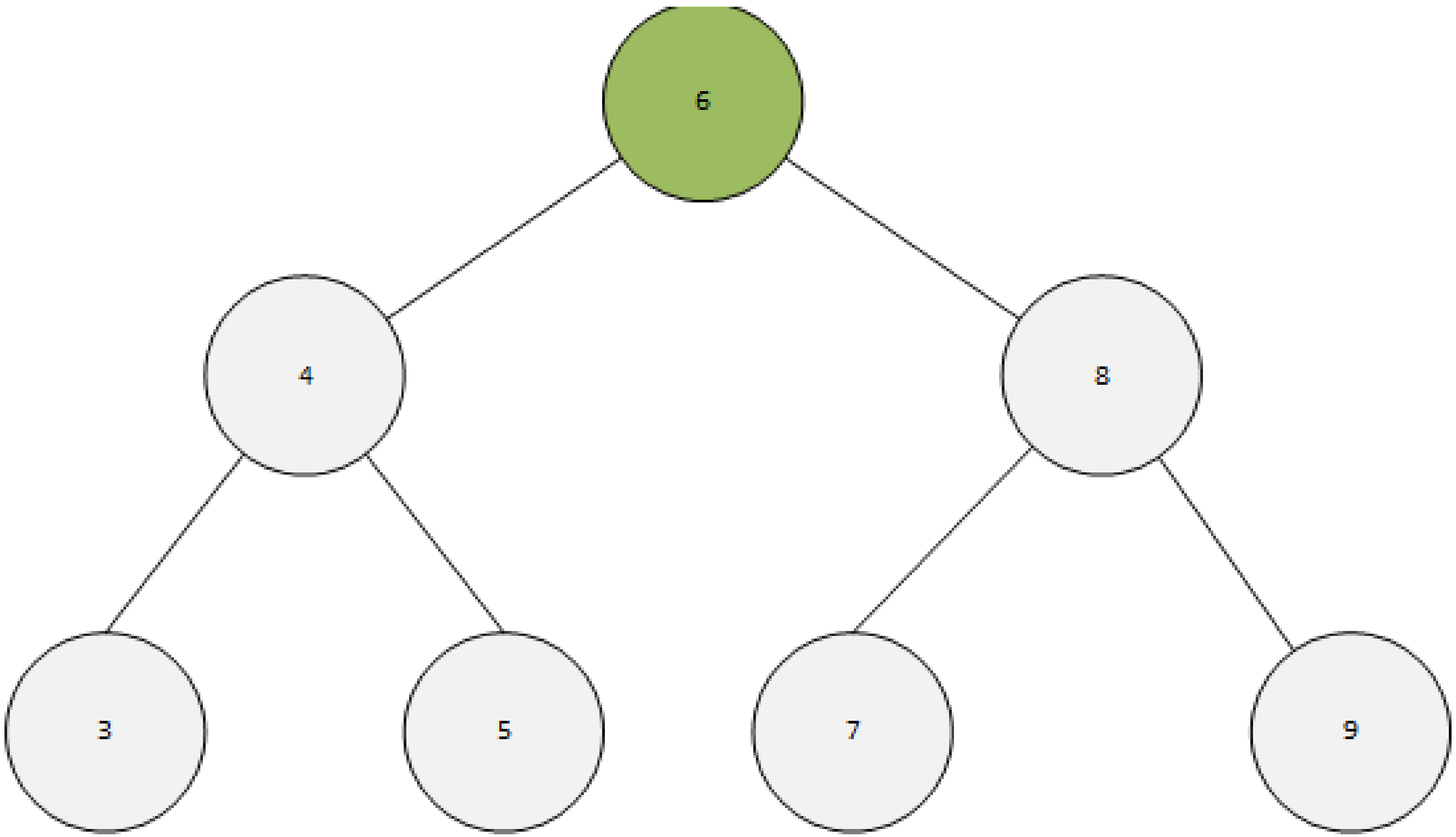


2. T1 is a full binary tree because each of its internal vertices has two children.
3. T2 is a full 3-ary tree because each of its internal vertices has three children.
4. In T3 each internal vertex has five children, so T3 is a full 5-ary tree.
5. T4 is not a full m-ary tree for any m because some of its internal vertices have two children and others have three children. But it is a 3-ary tree (not full).

Ordered Rooted Tree

1. An ordered rooted tree is a rooted tree where the children of each internal vertex are ordered.
2. **What is the ordering here?**

- **Left child of each internal vertex < internal vertex**
- **Right child of each internal vertex > internal vertex**

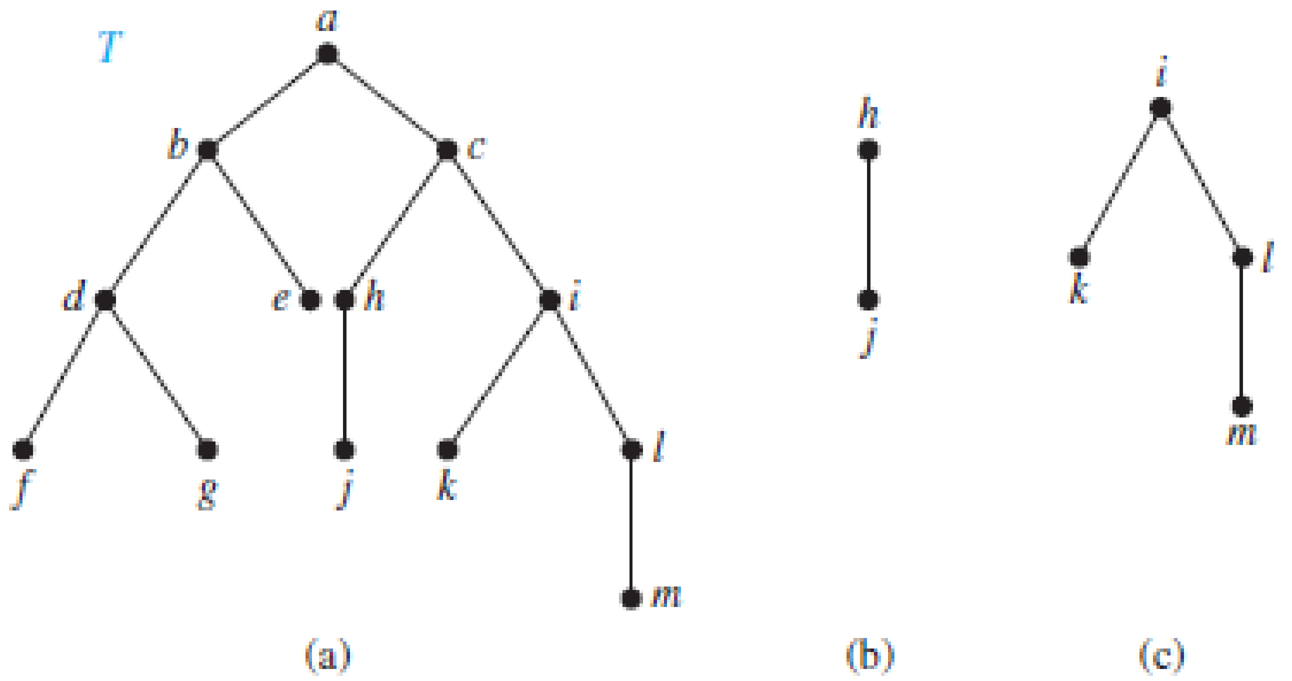


Ordered Rooted Tree

1. 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.
2. The tree rooted at the left child of a vertex is called the left subtree of this vertex, and
3. The tree rooted at the right child of a vertex is called the right subtree of the vertex

Ordered Rooted Tree

1. Example: What are the left and right children of d in the binary tree T shown below? (where the order is that implied by the drawing)?
2. What are the left and right subtrees of c?



Properties of Trees

1. **A tree with n vertices has n – 1 edges.**
   - We will prove this after we complete induction and recursion.
2. **A full m-ary tree with i internal vertices contains n = mi + 1 vertices**.
3. **Proof**: Every vertex, except the root, is the child of an internal vertex.
   - Because each of the i internal vertices has m children, there are mi vertices in the tree other than the root.
   - Therefore, the tree contains n = mi + 1 vertices.

Properties of Trees

1. A full m-ary tree with

n = number of vertices in the tree

i = number of internal vertices

l = number of leaves in this tree

Essentially, $n = i + l$

Once one of n, i, and l is known, the other two quantities are determined as:

- if n is known, $\mathbf{i = (n - 1)/m}$ internal vertices and $\mathbf{l = [(m - 1)n + 1]/m}$ leaves,
- If i is known, $\mathbf{n = mi + 1}$ vertices and $\mathbf{l = (m - 1)i + 1}$ leaves,
- If l is known, $\mathbf{n = (ml - 1)/(m - 1)}$ vertices and $\mathbf{i = (l - 1)/(m - 1)}$ internal vertices.

- You can derive this from the previous theorem.

Properties of Trees

1. **Example**: Suppose that someone starts a chain letter.
2. Each person who receives the letter is asked to send it on to **four** other people.
3. Some people do this, but others do not send any letters.
4. How many people have seen the letter, including the first person, if no one receives more than one letter and if the chain letter ends after there have been 100 people who read it but did not send it out?
5. How many people sent out the letter?

Properties of Trees

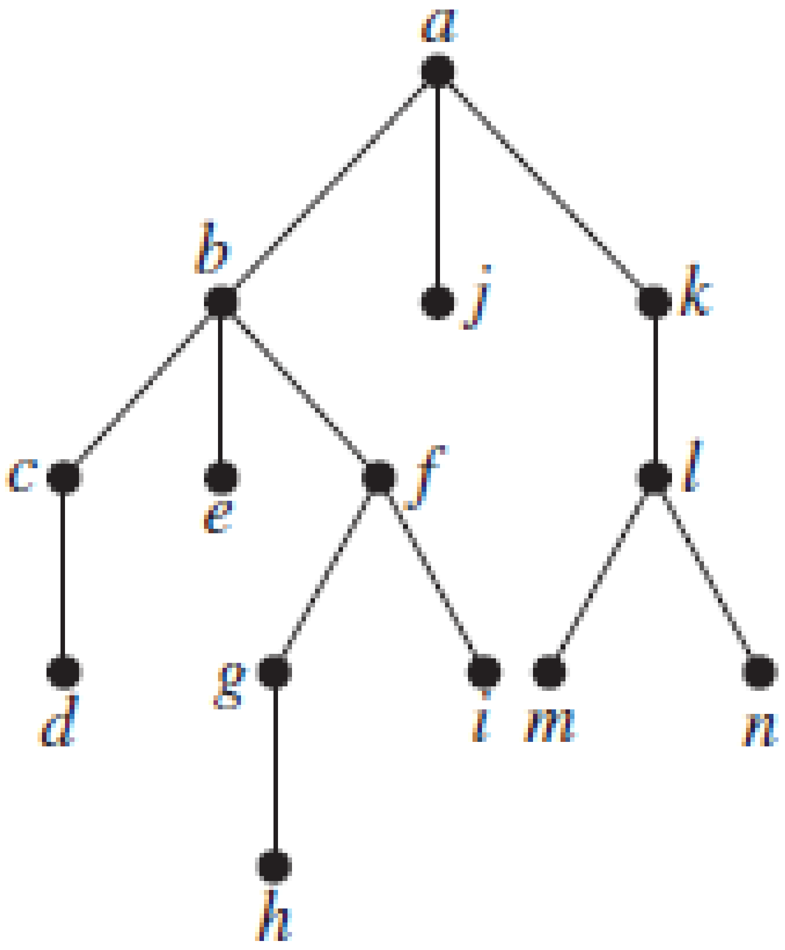
1. **Solution:** The chain letter can be represented using a 4-ary tree.
2. The internal vertices correspond to people who sent out the letter.
3. The leaves correspond to people who did not send it out.
4. Because 100 people did not send out the letter, the number of leaves in this rooted tree is $l = 100$.
5. Hence, part (iii) of the previous theorem, If l is known, $\mathbf{n = (ml - 1)/(m - 1)}$ vertices and $\mathbf{i = (l - 1)/(m - 1)}$ internal vertices.
6. The number of people who have seen the letter is $n = (4 \cdot 100 - 1)/(4 - 1) = 133$. [use the formula]
7. Also, the number of internal vertices is $133 - 100 = 33$ [$i = n - l$]
8. So 33 people sent out the letter.

Balanced m-ary Trees

1. Rooted trees that are “balanced” have the property that the subtrees at each vertex contain paths of approximately the same length.
2. To understand this concept, the following terminology will be used:
3. The **level** of a vertex v in a rooted tree is the length of the unique path from the root to this vertex.
4. The level of the root is defined to be zero.
5. The **height** of a rooted tree is the maximum of the levels of vertices.

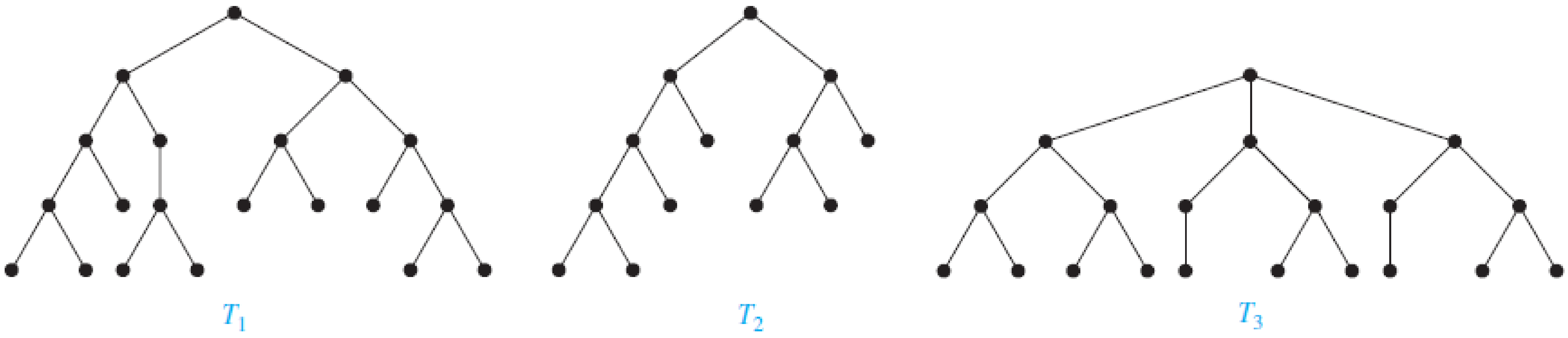
Balanced m-ary Trees

1. Example: Find the level of each vertex in the rooted tree. What is the height of this tree?
2. Solution: The root a is at level 0.
3. Vertices b, j , and k are at level 1.
4. Vertices c, e, f , and l are at level 2.
5. Vertices d, g, i, m, and n are at level 3.
6. Vertex h is at level 4.
7. Because the maximum level of any vertex is 4, this tree has height 4.



Balanced m-ary Trees

1. **Definition: A rooted m-ary tree of height h is balanced if all leaves are at levels h or h − 1.**
2. Example: Which of the rooted trees shown below are balanced?



3. Solution: T1 is balanced, because all its leaves are at levels 3 and 4.
4. T2 is not balanced, because it has leaves at levels 2, 3, and 4.
5. T3 is balanced, because all its leaves are at level 3.

m-ary Tree – Upper bound on number of leaves

1. **Theorem: There are at most m^h leaves in an m-ary tree of height h**.
    - We will prove this after we complete induction and recursion

2. Corollary: If an m-ary tree of height h has l leaves, then $h \geq \lceil \log_m l \rceil$

3. Corollary: If the m-ary tree is full and balanced, t $h = \lceil \log_m l \rceil$