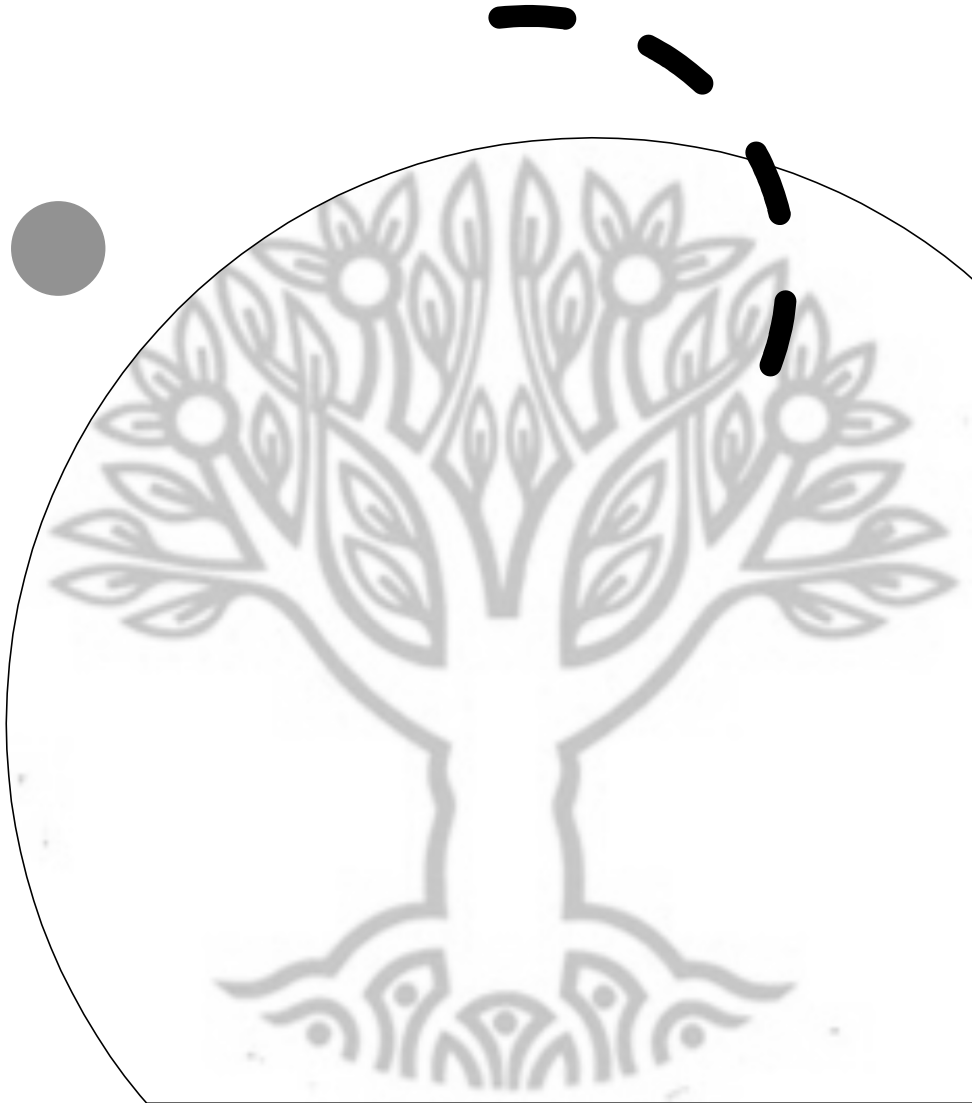
Trees



Lesson 2.1

TREES

The Simplest Level of Life is Most Powerful

Wholeness of the lecture: A tree is a very simple graph—it is connected but has no cycles. Trees have applications in communication networks, data structures, sorting, data compression, and routing algorithms. *Science of Consciousness:* The simplest level of life is the most powerful. The Transcendental Meditation technique gives us access to the simplest level of our own awareness.

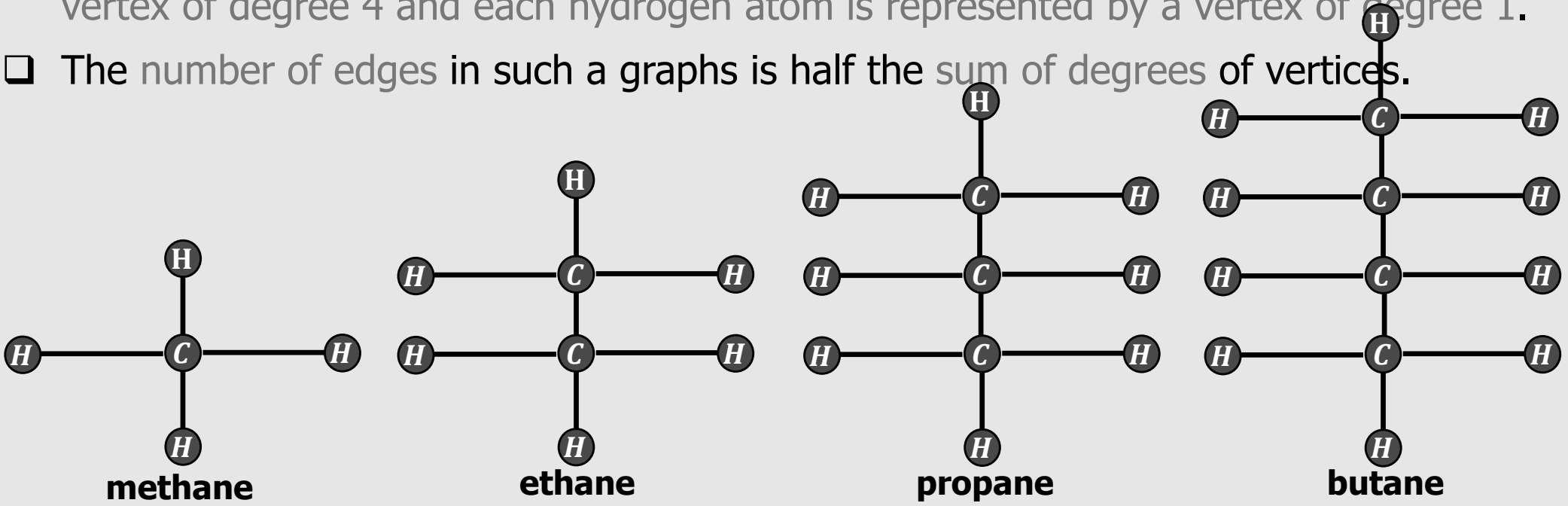
Main Points

1. A tree is a connected graph without any cycles. A tree has the minimum number of edges needed to connect all its vertices. *Science of Consciousness:* Practicing the Transcendental Meditation technique leads to efficiency in action: do less and accomplish more.[4]
2. A tree traversal represents a tree as a string of vertices that contains all the information in the tree but can be more easily used by a computer. *Science of Consciousness:* Knowledge has organizing power. By transcending, we contact the field of pure knowledge and bring infinite organizing power into our lives.

Trees

Saturated Hydrocarbons and Trees

- In 1857 Arthur Cayley studied hydrocarbons, chemical compounds from hydrogen and carbon atoms.
- In graph models of saturated hydrocarbons, each carbon atom is represented by vertex of degree 4 and each hydrogen atom is represented by a vertex of degree 1.
- The number of edges in such a graphs is half the sum of degrees of vertices.



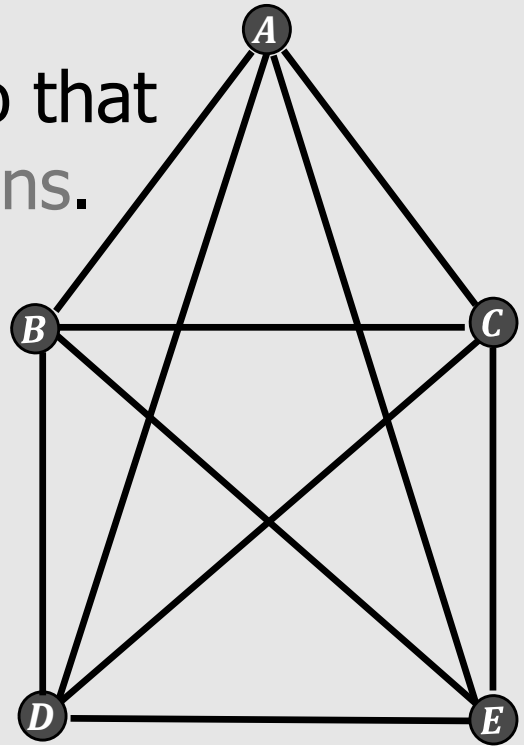
Trees

Telephone Network in a Town

- The following graph illustrates the telephone network between different towns.

- To save the cost and time build a few lines so that the route calls be possible through other towns.

- The edges {A, B}, {A, C}, {B, D} and {C, E} can be left out without disrupting communication between any two towns.



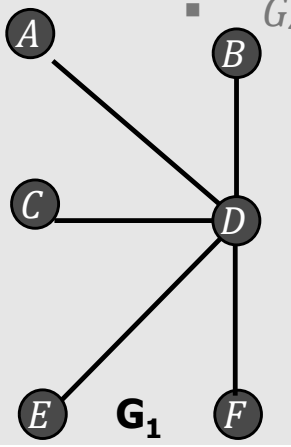
Trees

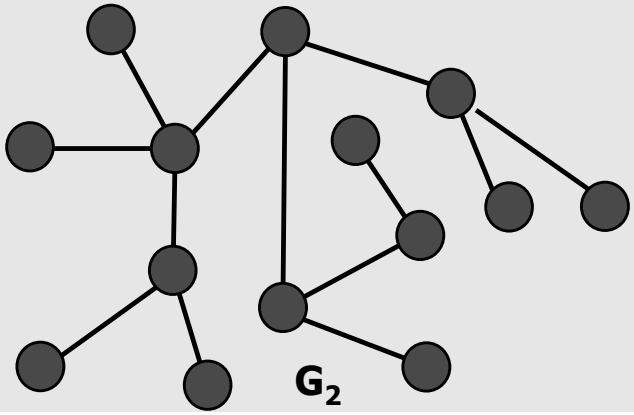
Tree

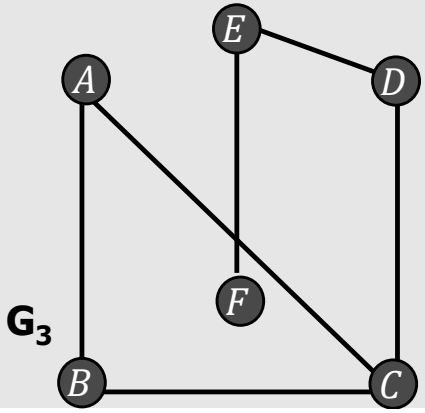
- A **tree** is a connected undirected graph in which every **pair of vertices** have **exactly one simple path** between them.

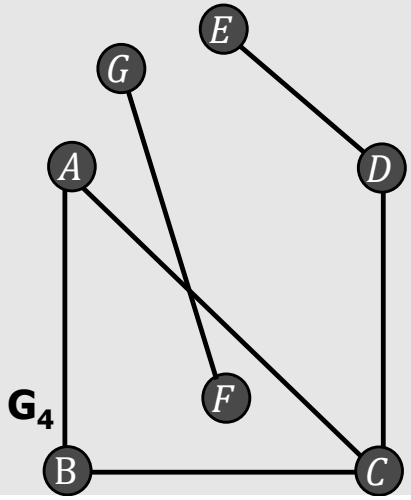
OR

- A **tree** is a connected undirected graph having **no simple circuits**.
- A tree can not be simple circuit, a tree can not contain multiple edges or loops.
  - **Example:**
    - Since G_1 and G_2 are trees because these are undirected graphs with no simple circuits.
    - G_3 is not a tree because ABCA is a simple circuit in it.
    - G_4 is not a tree because this is not a connected graph.





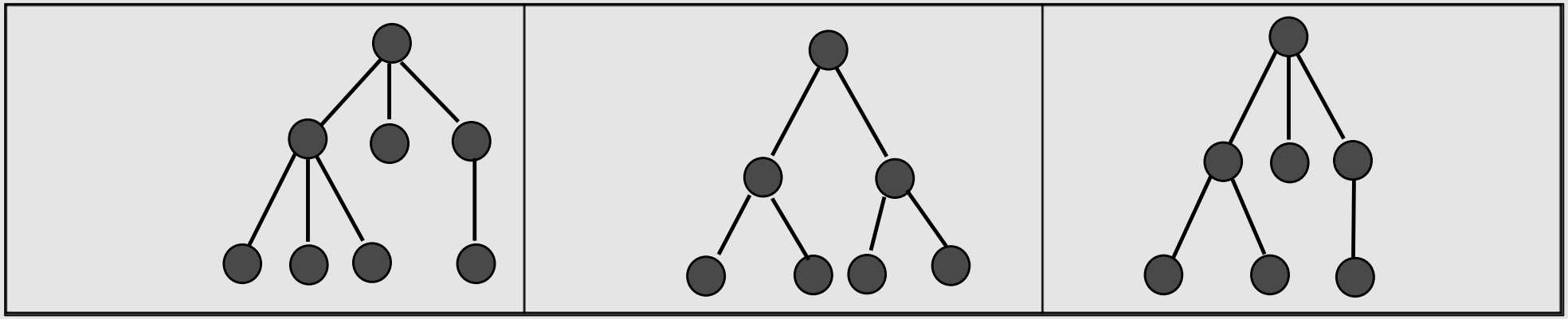




Trees

Forest

- A graph who's each connected component is a tree is called forest. A forest is a graph that has no simple circuit but is not connected.
- Graphs containing no simple circuits that are not connected, but each connected component is a tree.



Trees

Trees as Model

- Trees can be widely used in the field of computer science, chemistry, geology, botany and some other fields.
  - **Some important models based on trees**
    - Saturated Hydrocarbons and Trees
    - Representing Organizations
    - An Organizational Tree for a Computer Company
    - Computer File Systems
    - Tree-Connected Parallel Processors

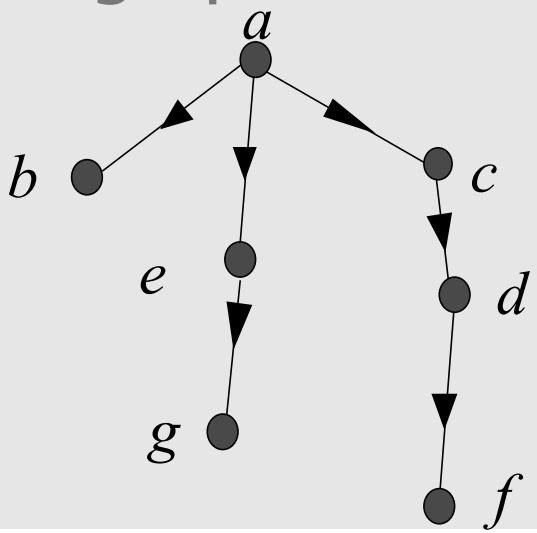
Trees

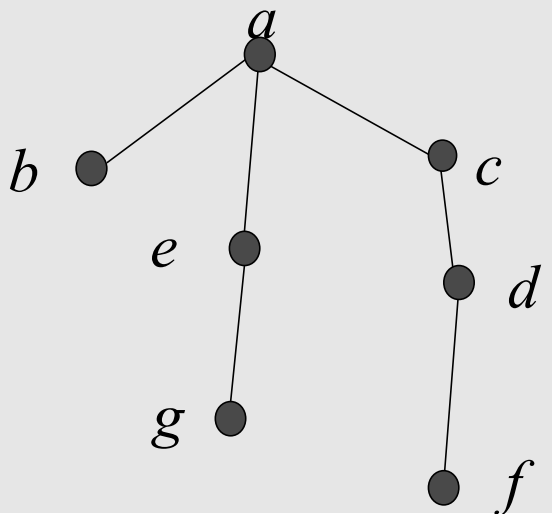
Rooted Tree Terminologies

- If v is a vertex of a rooted tree other than the root, the parent of v is the unique vertex u such that there is a directed edge from u to v.
- When u is a parent of v, v is called a child of u. Vertices with the same parent are called siblings.
- The ancestors of a vertex are the vertices in the path from the root to this vertex, excluding the vertex itself and including the root.
- The descendants of a vertex v are those vertices that have u as an ancestor.

Trees

- A vertex of a rooted tree with no children is called a leaf.
- Vertices that have children are called internal vertices.
- If v is vertex in a tree, the subtree with v as its root is the subgraph of the tree consisting of vertex v.

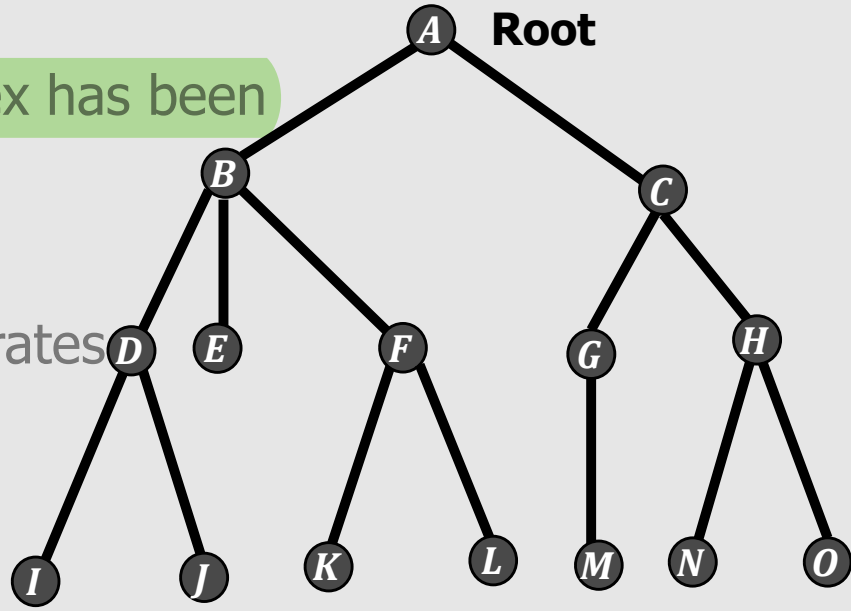


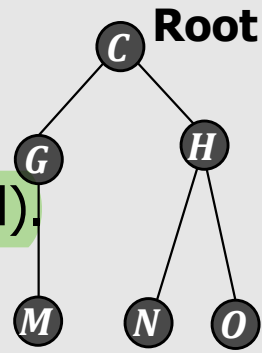


Trees

Rooted Tree

- A rooted tree is a type of tree in which one vertex has been designated as root.
- Every edge is directed away from the root.
  - **Note**
    - A rooted tree with n-different vertices generates n-different rooted trees i.e., when different vertices are chosen as the root.
      - **Example**
        - ✓ The *A* is a parent of *B*.
        - ✓ The children of *B* are *D*, *E* and *F*.
        - ✓ The ancestors of *F* are *B* and *A*.
        - ✓ The descendants of *C* are *G*, *H*, *M*, *N* and *O*.
        - ✓ The siblings of *G* is *H*.
        - ✓ The internal vertices are *A*, *B*, *C*, *D*, *F*, *G* and *H* (at least one child).
        - ✓ The leaves are *E*, *I*, *J*, *K*, *L*, *M*, *N* and *O* (deg=1).
        - ✓ At the end subtree with *C* as its root.

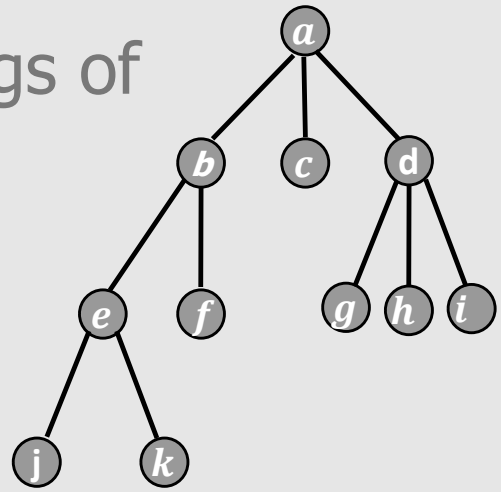


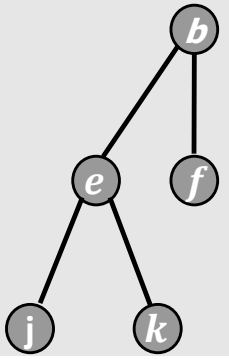


Trees

Example

- In the given rooted tree (with root a):
  Find the parent of e, the children of d, the siblings of g, the ancestors of k, and the descendants of b?
  Find all internal vertices and all leaves?
  Find the subtree rooted at b?
  - **Solution:**
    - The parent of e is b.
    - The children of d are g, h, and i.
    - The siblings of g are h and i.
    - The ancestors of k are e, b and a.
    - The descendants of b are e, j, k and f.
    - The internal vertices are a, b, d, and e.
    - We can display the subtree rooted at b.





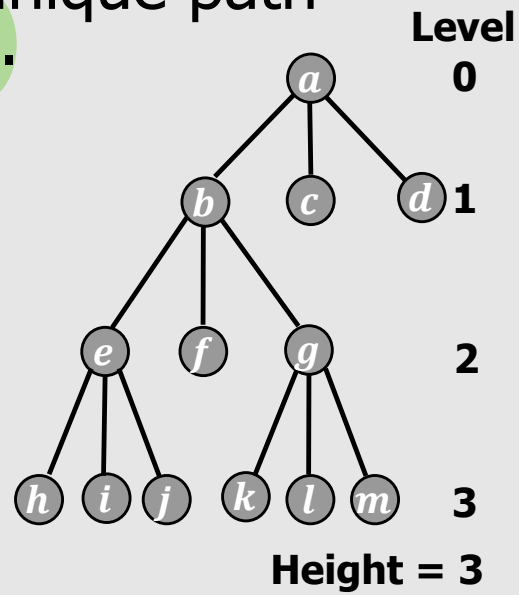
Trees

Level

- The level of a vertex v in a rooted tree is the length of unique path from the root to vertex.

Height

- The height of a rooted tree is the sum of all the levels of the vertices.
- The height of the tree is the maximum length of unique path joining root with the leaves. Length of root is zero.
  - **Example**
    - Find the level of each vertex in the rooted tree?
    - What is the height of the tree?
  - **Solution**
    - The root a is at level 0. Vertices b, c and d are at level 1.
    - Vertices e, f, g at 2 and h, i, j, k, l, and m are at level 3. The height of the tree is 3.



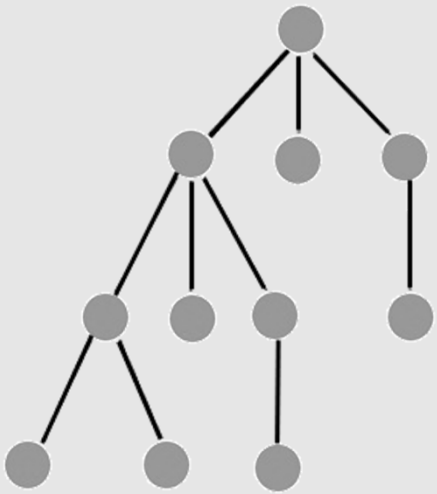
Trees

n - ary Tree

- A rooted tree is called n-ary tree if every vertex has no more than n children.

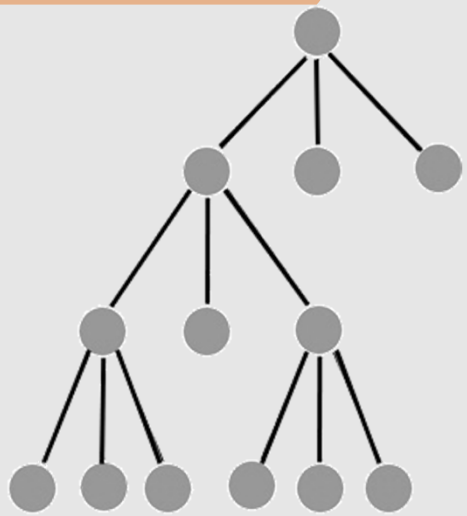
Full n-ary Tree

- It is called full n-ary tree if every internal vertex has exactly n children.
- A 2-ary tree is called binary tree.



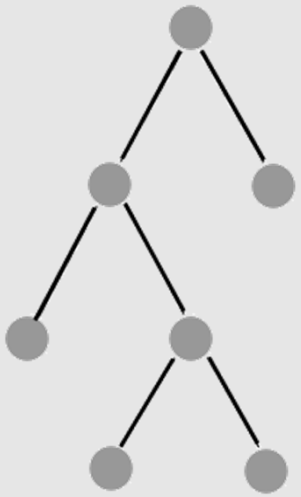
3-Ary Tree

Each internal vertex has no more than 3 children



Full 3-Ary Tree

Each internal vertex has exactly 3 children



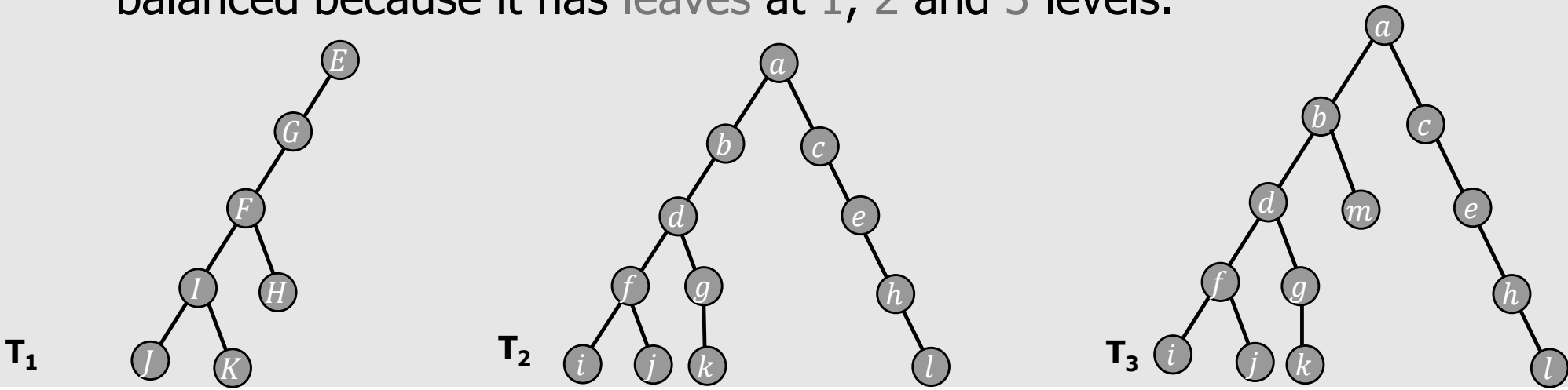
Full Binary Tree

Each internal vertex has exactly 2 children

Trees

Balanced n-Ary trees

- A rooted n-Ary tree of height h is balanced if all leaves are at the levels of h or h – 1.
  - Example: Find the balanced rooted trees?
  - Solution:
    - Rooted trees T_1 and T_2 are balanced rooted trees and T_3 is not balanced because it has leaves at 1, 2 and 3 levels.



Trees

Ordered Rooted Trees

- An ordered rooted tree is a rooted tree where the children of each internal vertex are ordered.
- We draw ordered rooted trees so that the children of each internal vertex are shown in order from left to right.

Binary Ordered Rooted Tree

- A binary tree is an ordered rooted tree in which each vertex has at most two children, the first child is called the left child and the second one is the right child.
- The tree rooted at the left child of a vertex is called left subtree of this vertex, and the tree rooted at the right child of a vertex is called the right subtree of this vertex.

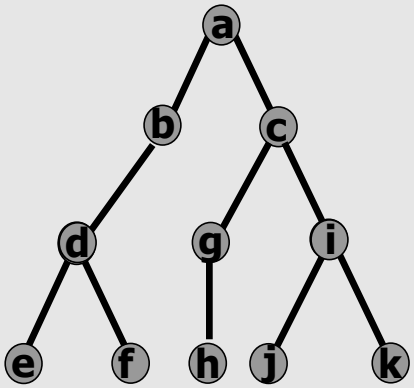
Trees

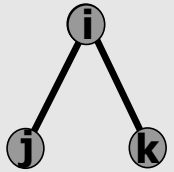
Example

- Consider the binary tree T.
- What are the left and right children of d?
- What are the left and right subtrees of C?

➢ **Solution**

- The left child of d is e and right child of d is f.
- The left subtree and right subtree of C are displayed.

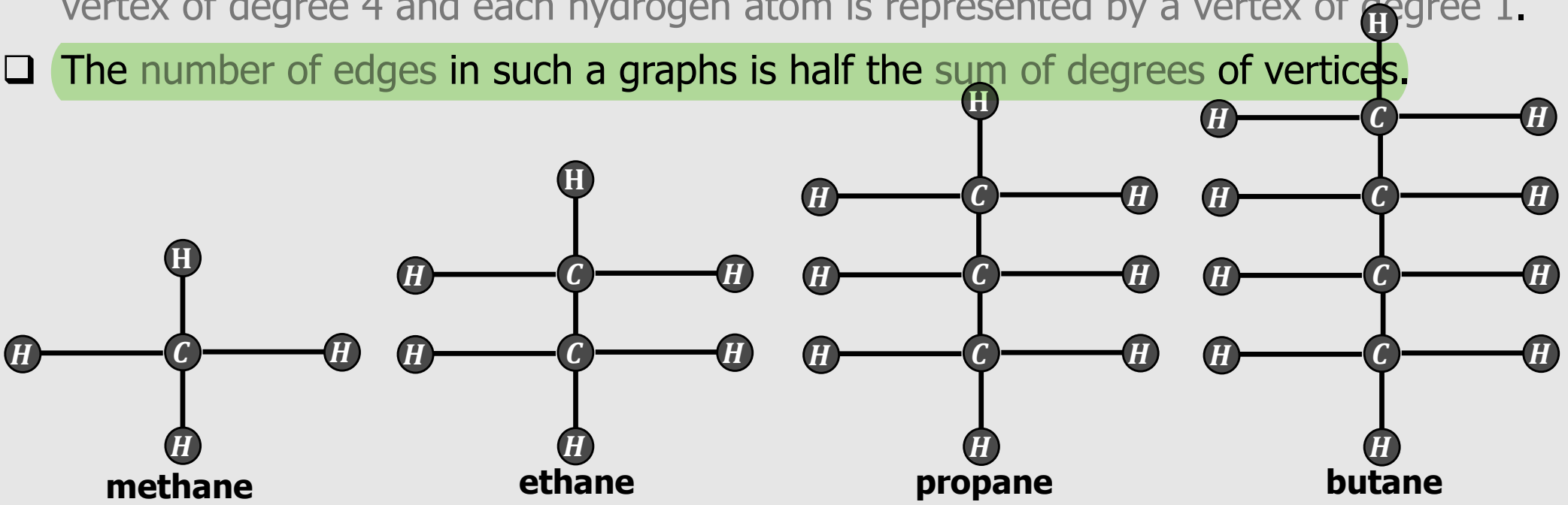




Trees

Saturated Hydrocarbons and Trees

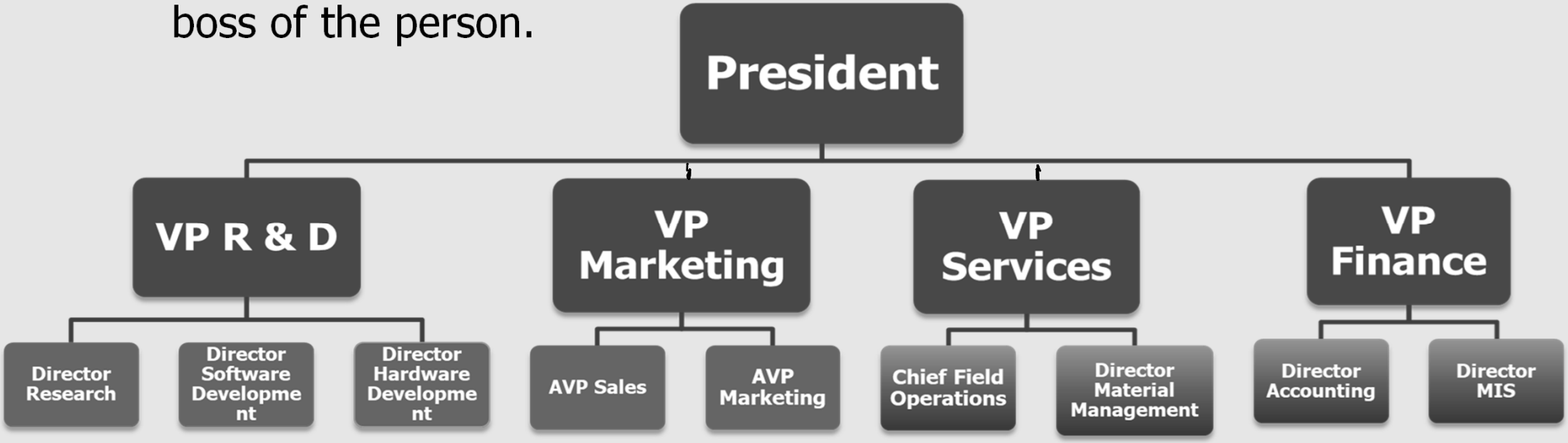
- In 1857 Arthur Cayley studied hydrocarbons, chemical compounds from hydrogen and carbon atoms.
- In graph models of saturated hydrocarbons, each carbon atom is represented by vertex of degree 4 and each hydrogen atom is represented by a vertex of degree 1.
- The number of edges in such a graphs is half the sum of degrees of vertices.



Trees

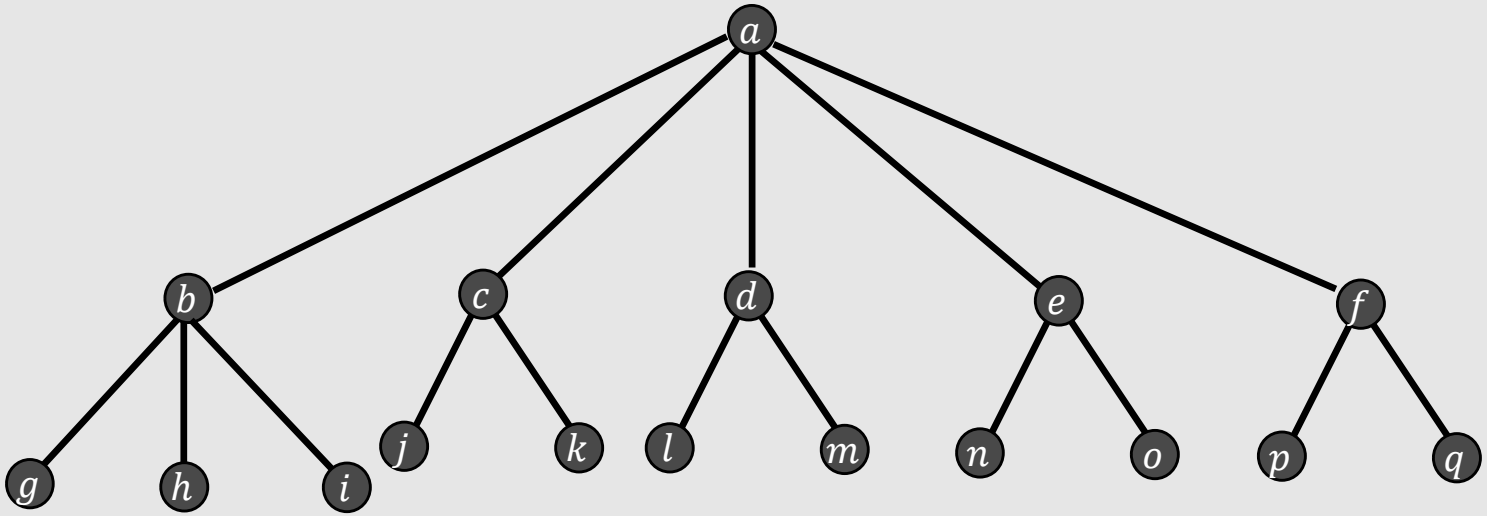
Trees as Models

- **An Organizational Tree for a Computer Company:**
  - An administrative structure of big computer company can be modeled using a rooted tree.
  - Each vertex in this tree represents a position in the organization.
  - An edge from one vertex to another vertex representing the immediate boss of the person.



Trees

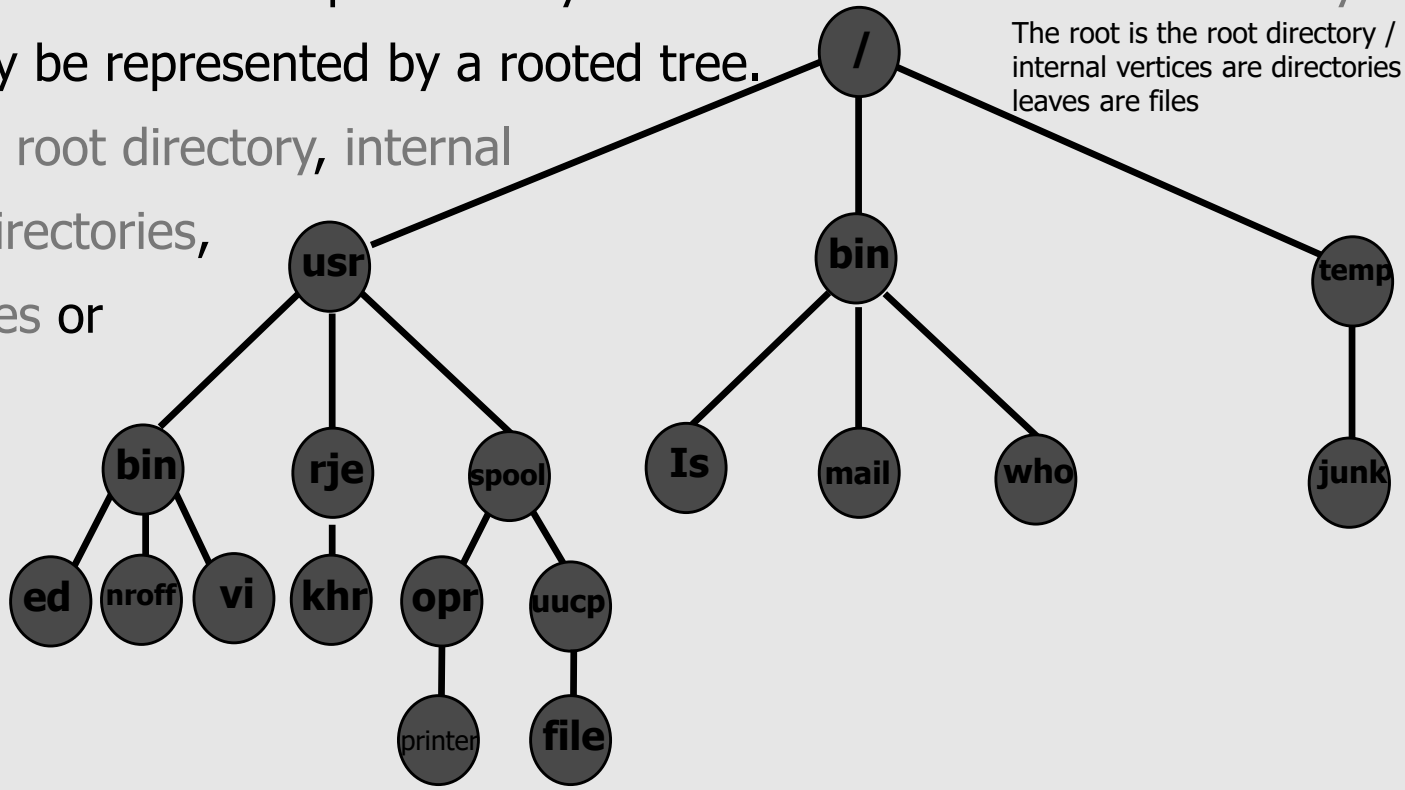
Rooted Tree diagram for a Compute Company



Trees

An Organizational Tree for a Computer Company

- Files in the computer memory are organized into directories. A directory can contain both files and subdirectories. The complete file system is stored in the root directory.
- Hence a file system may be represented by a rooted tree.
- The root represents the root directory, internal vertices represent subdirectories, and leaves represent files or empty directories.



Tree Traversal

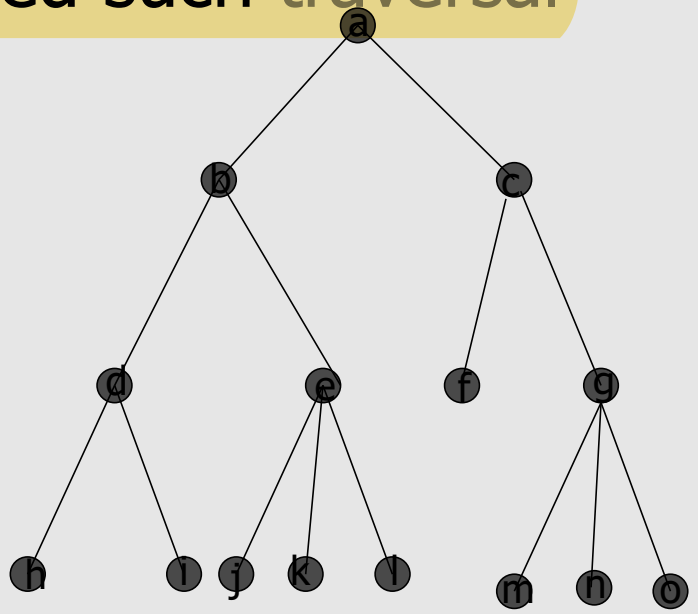
Introduction

- Ordered rooted trees can be used to store the information. To access the data someone, need to visit every vertex of an ordered rooted tree.
- There are different types of rooted trees by visiting each edge in a different order.
- The ordered rooted trees can also be used to represent various types of expressions like as arithmetic expressions involving parenthesis, operations, numbers and variables.
  - **Example**
    - The expressions of the form $3 \div (2 + x) - (4 + 3 \times 2)$ are necessary to be performed before the execution of computer program.
    - To solve such kind of arithmetic operations we use binary tree created by the operations as internal vertices and operands as terminal vertices is called an expression tree.

Tree Traversal

Traversal Algorithms

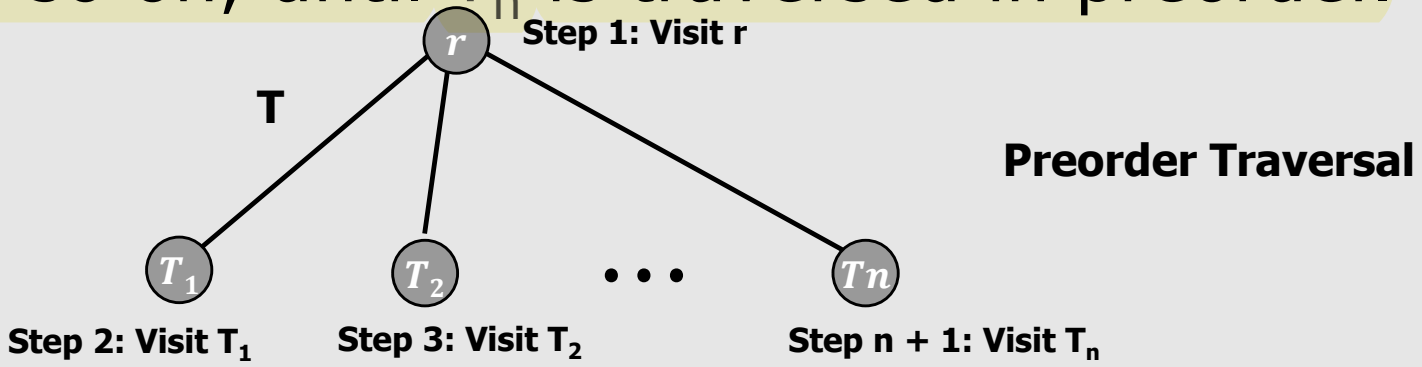
- A tree traversal requires that each vertex of the tree be visited exactly once in a predetermined sequence.
- We will learn three of the most used such traversal algorithms,
  i. Preorder traversal,
  ii. Inorder traversal, and
  iii. Postorder traversal



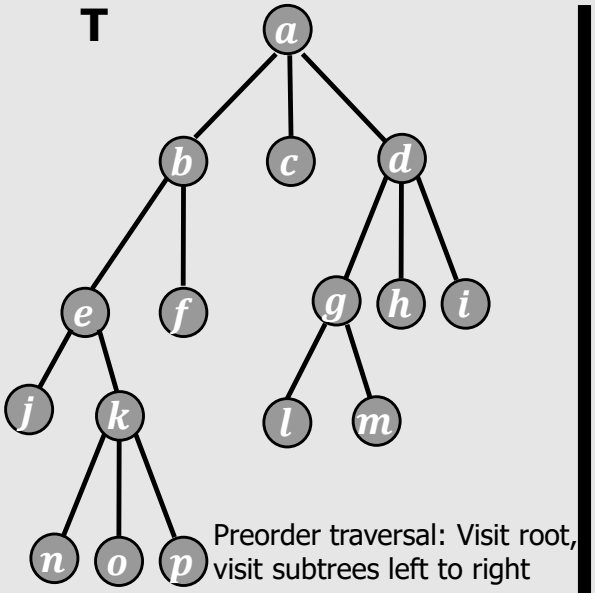
Tree Traversal

Preorder traversal

- Let T be an ordered rooted tree with root r. If T consists only of r, then r is the preorder traversal of T.
- Otherwise, suppose that T_1, T_2, T_3, …, T_n are the subtrees at r from left to right in T.
- The preorder traversal **begins by visiting r**.
- It continues by traversing T_1 in preorder, then T_2 in preorder, and so on, until T_n is traversed in preorder.

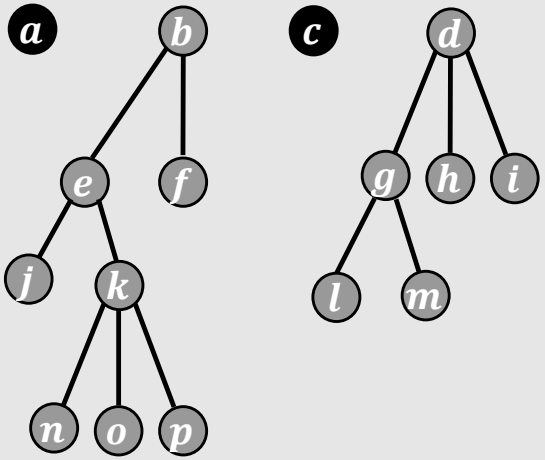


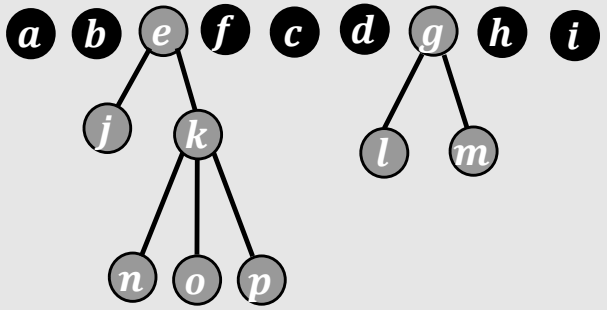
Tree Traversal

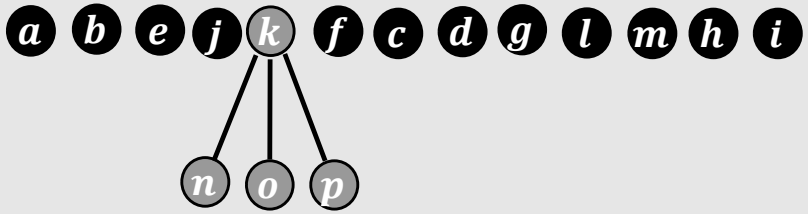


Preorder traversal: Visit root, visit subtrees left to right

The ordered rooted tree T





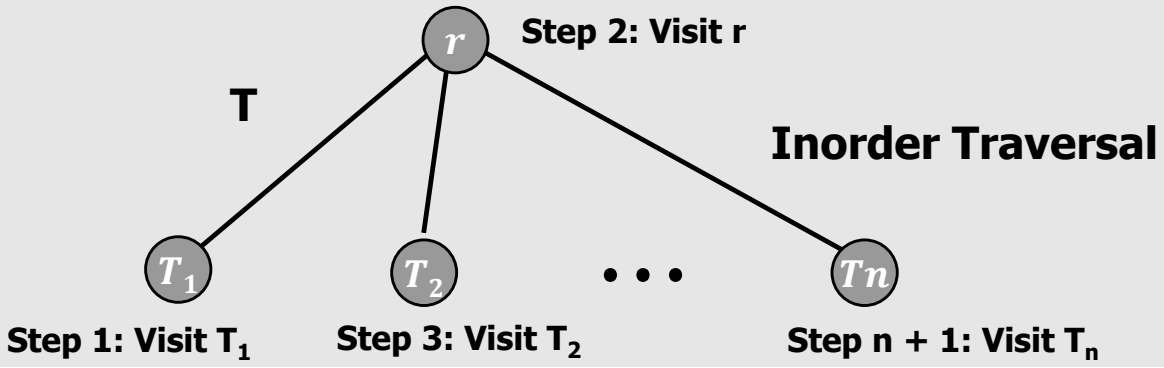


The Preorder Traversal of T

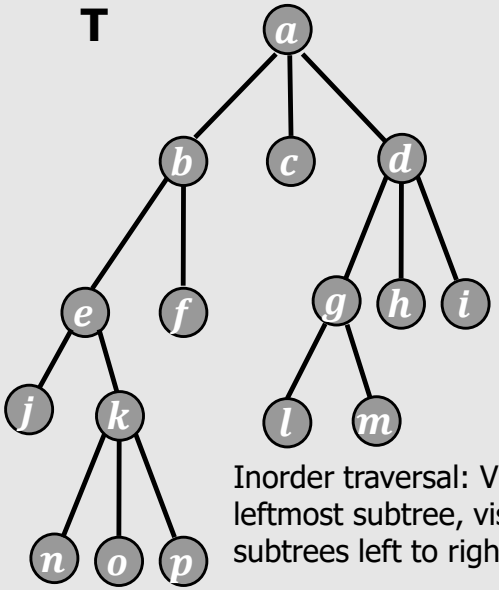
Tree Traversal

Inorder traversal

- Let T be an ordered rooted tree with root r.
- If T consists only of r, then r is the inorder traversal of T.
- Otherwise, suppose that T_1, T_2, T_3, …, T_n are the subtrees at r from left to right in T.
- The inorder traversal **begins by visiting T_1 inorder, then visiting r**.
- It continues by traversing T_2 in inorder, then T_3 in inorder, and so on, until T_n is traversed in inorder.

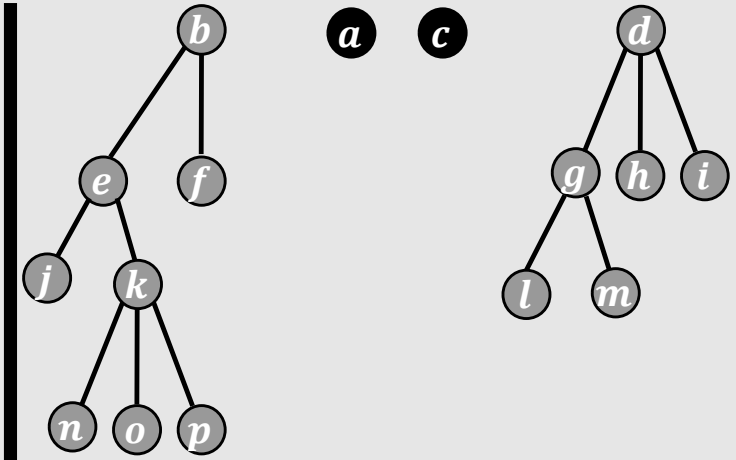


Tree Traversal

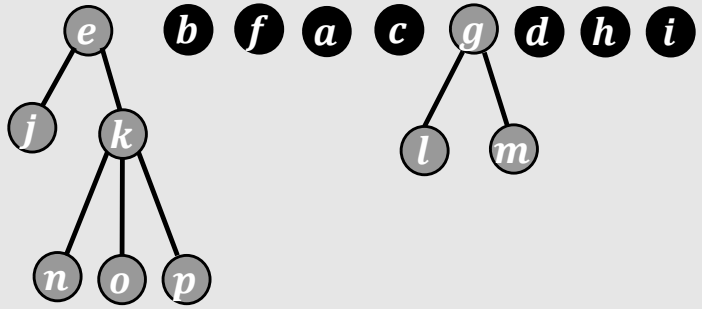


Inorder traversal: Visit leftmost subtree, visit subtrees left to right

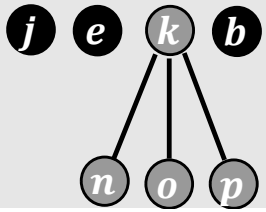
The ordered rooted tree T



a c



b f a c g d h i



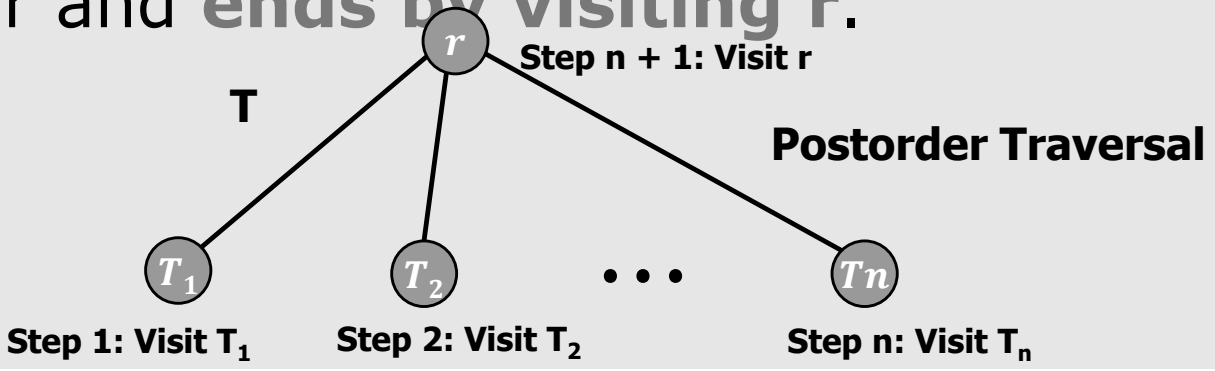
j e k b f a c l g m d h i

j e n k o p b f a c l g m d h i

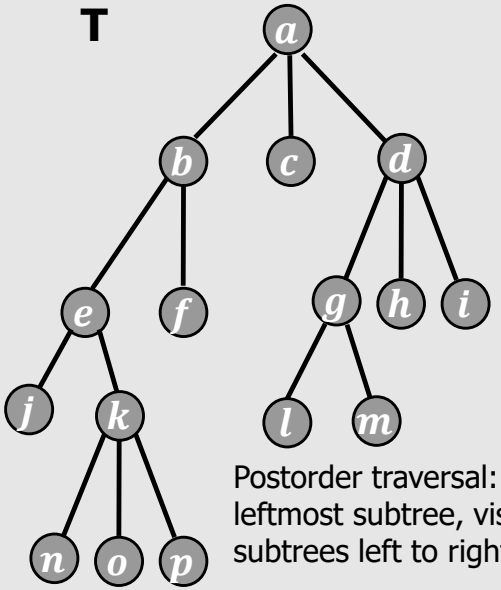
The Inorder Traversal of T

Trees Traversal

1. **Postorder traversal:**
   - Let T be an ordered rooted tree with root r.
   - If T consists only of r, then r is the postorder traversal of T.
   - Otherwise, suppose that T_1, T_2, T_3, …, T_n are the subtrees at r from left to right in T.
   - The postorder traversal **begins by visiting T_1** in postorder, then T_2 in inorder, and so on, until T_n is traversed in postorder and **ends by visiting r**.

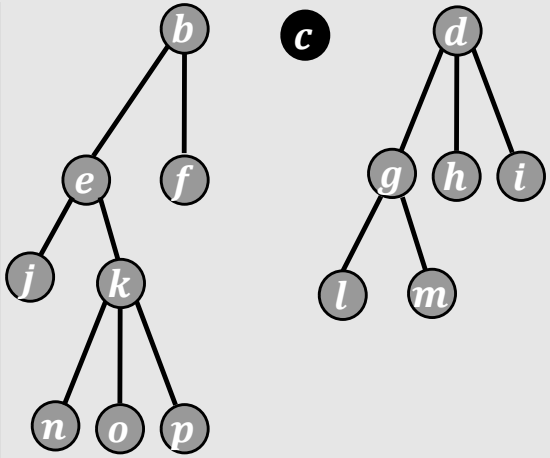


Trees Traversal



Postorder traversal: Visit leftmost subtree, visit subtrees left to right

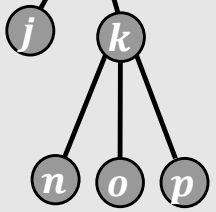
The ordered rooted tree T



c

a

e f b c g h i d a



j k e f b c l m g h i d a

j n o p k e f b c l l m g h i d d

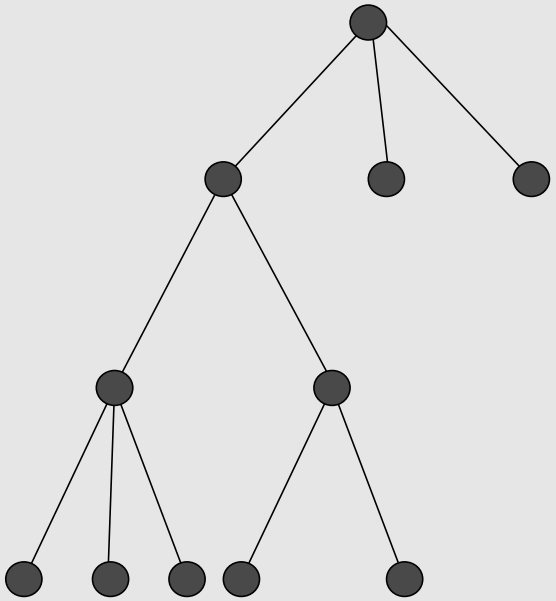
The Postorder Traversal of T

Properties of Trees

- **Properties of Trees:**
  - **i. Theorem-1:**
    - An undirected graph is a tree if and only if between every pair of vertices there is exactly one simple path.
  - **ii. Theorem-2:**
    - A tree with n vertices has $n-1$ edges.
  - **iii. Theorem-3:**
    - A full $m-ary$ tree with i internal vertices contains $n = mi + 1$ vertices.
  - **iv. Theorem-4:**
    - A full $m-ary$ tree with
      - a. n vertices has $i = \frac{n-1}{m}$ internal vertices and $l = \frac{1}{m}[(m-1)n+1]$ leaves,
      - *b.* i internal vertices has $n = mi + 1$ vertices and $l = (m-1)i + 1$ leaves,
      - *c.* l leaves has $n = \frac{ml-1}{m-1}$ vertices and $i = \frac{l-1}{m-1}$ internal vertices.
  - **v. Theorem-5:**
    - There are at the most m^n leaves in a $m-ary$ tree of height n.

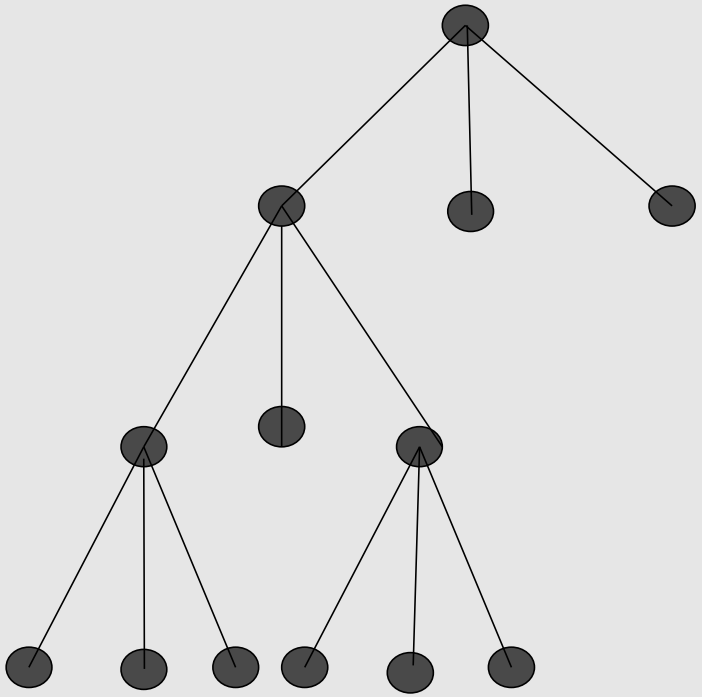
Properties of Trees

- **Properties of Trees:**
  - i. **Theorem-1:**
    - A tree with n vertices has $n - 1$ edges.



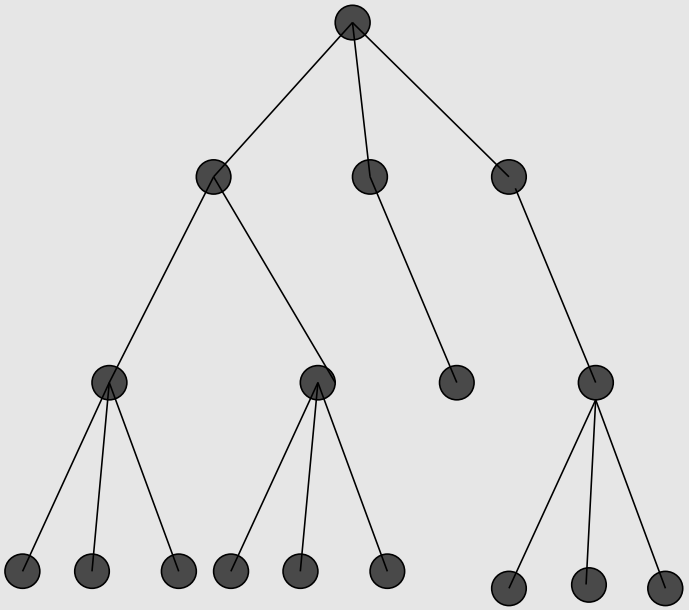
Properties of Trees

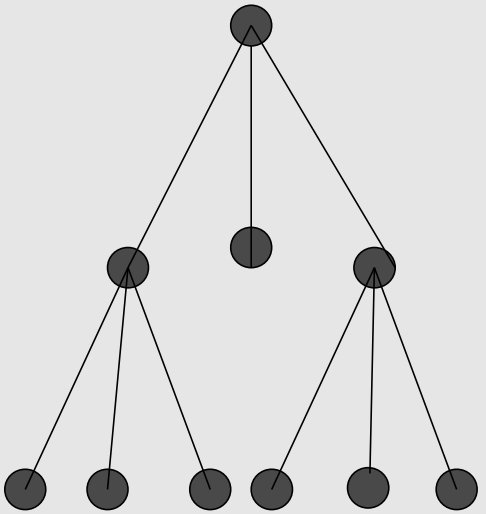
- **Properties of Trees:**
  - i. **Theorem-3:**
    - A full $m-ary$ tree with i internal vertices contains $n = mi + 1$ vertices.



Properties of Trees

- **Properties of Trees:**
  - i. **Theorem-5:**
    - There are at the most m^n leaves in a $m - ary$ tree of height n.





UNITY CHART

Connecting The Parts Of Knowledge With The Wholeness Of Knowledge

Trees

1. A tree is a very simple graph: it is connected and has no cycles.
2. A tree traversal gives a string of vertices that embodies the entire tree but is easier to use.

---

3. **Transcendental Consciousness** is a field of pure knowledge.
4. **Impulses within the transcendental field** have infinite organizing power.
5. **Wholeness moving within itself:** In Unity Consciousness, we use the infinite organizing power of own Self to handle all possibilities in life.

Applications of Trees

- **Binary Search Tree**
  - This is a recursive kind of procedure which is used to construct a binary search tree for a list of items. Start with a tree containing just one root. Binary search tree find items based on series of comparisons, where each comparison shows whether we have located the items or whether we should go right or left in a subtree.
- **Decision Tree**
  - Rooted trees are used to construct the problems in which series of decisions generates a solution. A rooted tree in which each internal vertex describes a decision, with a subtree at these vertices for each possible outcome of the decision is called a decision tree.
- **Prefix Codes**
  - A prefix code is constructed by any binary tree where the left edge at each internal vertex is labeled by 0 and the right edge by a 1 and where the leaves are labeled by characters.
- **Huffman Coding**
  - Huffman coding was developed by David Huffman to introduce an algorithm that takes as input the frequencies of symbols in a string and produces as output a prefix code that encodes the string using the fewest possible bits, among all possible binary prefix codes for these symbols.
- **Game Trees**
  - Game trees are used to study the games such as tic-tac-toe, nim, checkers, and chess. The models of these games use game trees.