Discrete Structures

Syed Faisal Bukhari, PhD

Associate Professor

Department of Data Science (DDS), Faculty of Computing and Information Technology (FCIT), University of the Punjab (PU)

Text book

Discrete Mathematics and Its Application, 7th Edition

Kenneth H. Rosen

References

Chapter 9

Discrete Mathematics and Its Application, 7th Edition

by Kenneth H. Rose

These slides contain material from the above resource.

TOURNAMENTS

- ❑ **Round-Robin Tournaments**

- ❑ **Single-Elimination Tournaments**

Round-Robin Tournaments

A tournament where each team **plays** every other team **exactly once** and **no ties are allowed** is called a **round-robin tournament**.

- Such tournaments can be modeled using **directed graphs** where **each team** is represented by **a vertex**.

- Note that ***(a, b)*** is an **edge** if team ***a* beats team *b***

- This graph is a **simple directed graph**, containing **no loops** or **multiple directed edges** (because no two teams play each other more than once).

Example of Round-Robin Tournaments

Such a **directed graph model** is presented in **Figure 13**. We see that **Team 1** is **undefeated** in this tournament, and **Team 3** is **winless**.

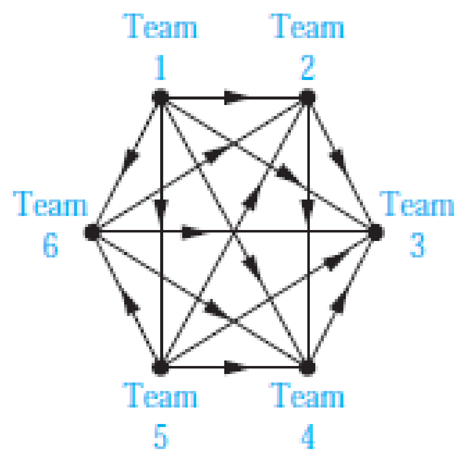


FIGURE 13 A Graph Model of a Round-Robin Tournament.

Single-Elimination Tournaments

- **Single-Elimination Tournaments** A tournament where each contestant is eliminated after one loss is called a **single-elimination tournament.**

- **Single-elimination tournaments** are often used in sports, including **tennis championships** and the yearly **NCAA basketball championship**.

- We can model such a tournament using a **vertex** to represent **each game** and **a directed edge** to connect a game to the next game the winner of this game played in.

Example of Single-Elimination Tournaments

The graph in **Figure 14** represents the games played by the final **16 teams** in the 2010 NCAA women's basketball tournament.

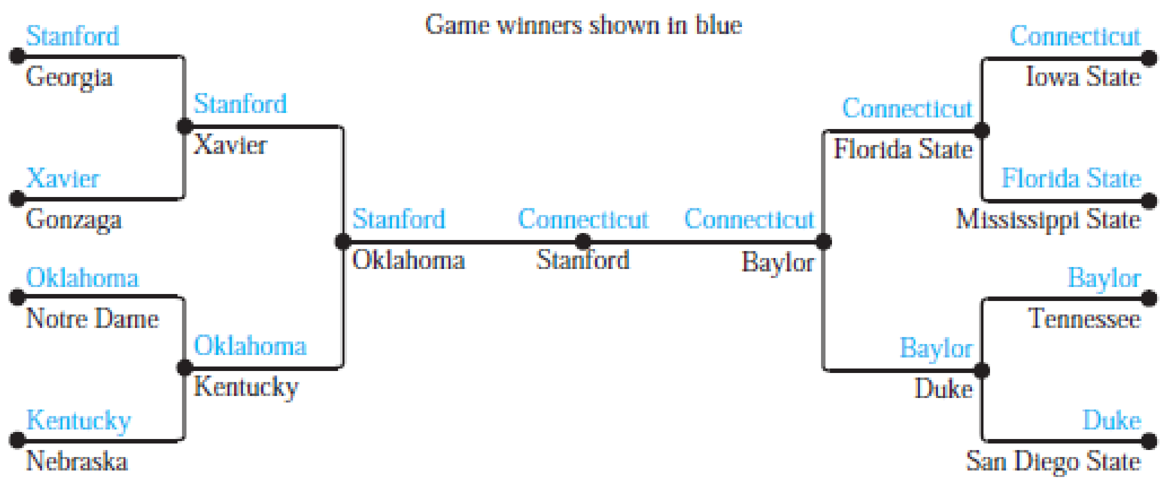


FIGURE 14 A Single-Elimination Tournament.

Adjacent (or Neighbors)

Definition: Two vertices **u** and **v** in an **undirected graph G** are called **adjacent (or neighbors)** in G if **u** and **v** are endpoints of **an edge e** of **G**. Such an **edge e** is called **incident** with the vertices **u** and **v** and **e** is said to connect **u** and **v**.

Definition: The **set of all neighbors** of a vertex **v** of G = (V, E), denoted by ***N*(v),** is called the **neighborhood** of **v**. If **A** is a subset of **V**, we denote by *N*(A) the set of all vertices in G that are adjacent to at least one vertex in A.

So, $N(A) = \bigcup_{v \in A} N(v)$

Degree of a Vertex

To keep track of how many edges are incident to a vertex, we use the following definition:

Definition: The **degree** of a vertex in an undirected graph is **the number of edges incident** with it, except **that a loop at a vertex contributes twice** to the degree of that vertex. The degree of the vertex v is denoted by deg(v).

Isolated vertex: a vertex of degree zero

Pendant vertex: a vertex of degree one

What are the **degrees** and what are the **neighborhoods** of the vertices in the graphs **G?**

deg(a) = 2

deg(b) = 4

deg(c) = 4

deg(d) = 1 **(Pendant vertex)**

deg(e) = 3

deg(f) = 4

deg(g) = 0 **(Isolated vertex)**

N(a) = {b, f}

N(b) = {a, c, e, f}

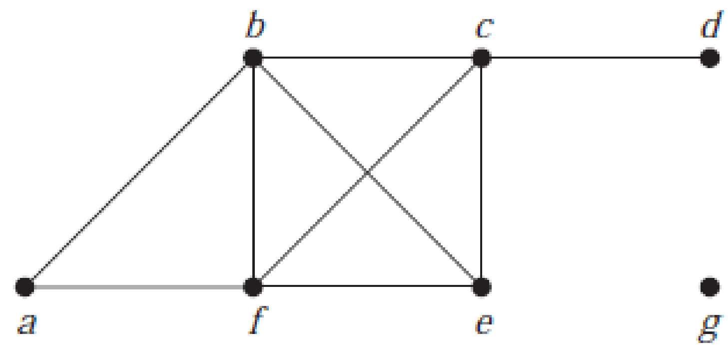
N(c) = {b, d, e, f}

N(d) = {c}

N(e) = {b, c, f}

N(f) = {a, b, c, e}

N(g) = {}



G

Example What are the **degrees** and what are the **neighborhoods of the vertices** in the graphs H

Solution

deg(a) = 4

deg(b) = 6

deg(c) = **1(Pendant vertex)**

deg(d) = 5

deg(e) = 6

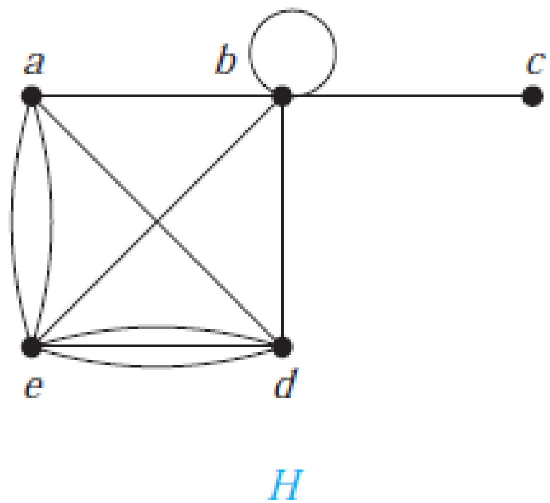
N(a) = {b, d, e}

N(b) = {a, b, c, d, e}

N(c) = {b}

N(d) = {a, b, e}

N(e) = {a, b, d}



THEOREM 1 THE HANDSHAKING THEOREM Let $G = (V, E)$ be an **undirected graph with *m* edges**. Then

$2m = \sum_{v \in V} deg(v)$

(Note that this applies even if **multiple edges** and **loops are present**.)

Example How many edges are there in a graph with 10 vertices each of degree six?

Solution:

Let $G = (V, E)$ be an undirected graph with ***m* edges**. Then

$2m = \sum_{v \in V} \deg(v)$

The sum of the degrees of the vertices is $6 \times 10 = 60$.

According to the **Handshaking theorem**

$2m = 60$ $\qquad$ $\because$ ***m* is the number of edges.**

$\Longrightarrow m = 30.$

THEOREM 1 shows that the **sum of the degrees of the vertices** of an **undirected graph is even**.

This simple fact has many consequences, one of which is given as Theorem 2.

THEOREM 2 An **undirected graph** has an **even number** of vertices of **odd degree**.

Proof: Let V_1 and V_2 be the set of **vertices of even degree** and the set of **vertices of odd degree**, respectively, in an undirected graph $G = (V, E)$ with m edges. Then

$$2m = \sum_{v \in V} deg(v) = \sum_{v \in V_1} deg(v) + \sum_{v \in V_2} deg(v)$$

Because **$deg(v)$ is even for $v \in V_1$**, the **first term in the right-hand** side of the **last equality is even**.

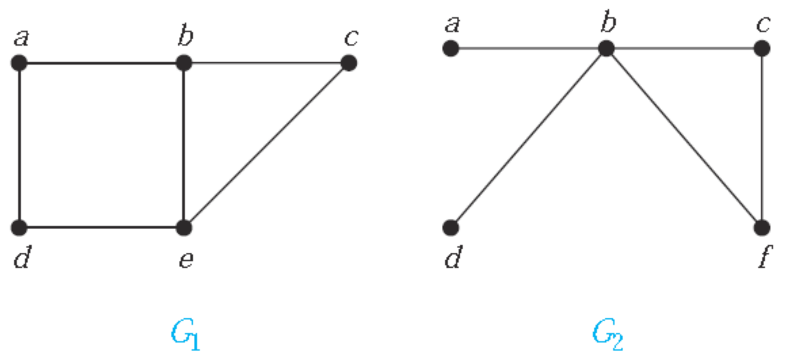
Furthermore, the sum of the **two terms on the right-hand** side of the **last equality is even**, because this sum is $2m$.

Hence, the **second term** in the **sum is also even**. Because all the terms in **this sum are odd**, there must be **an even number of such terms**.

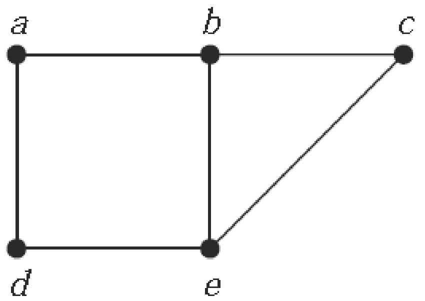
Thus, there are an **even number of vertices of odd degrees**.

Union of Graphs

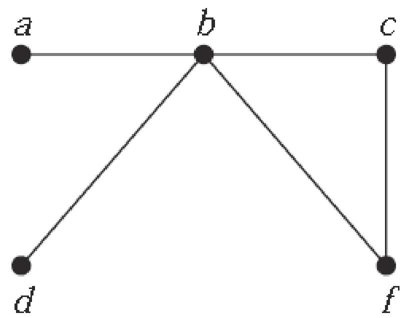
Example: Find the union of the given pair of simple graphs. (Assume edges with the same endpoints are the same.)



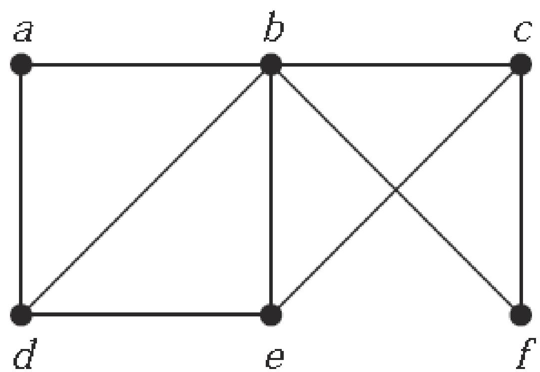
Solution



G_1



G_2



$G_1 \cup G_2$

Traversal Algorithms

Procedures for **systematically visiting** every vertex of an ordered rooted tree are called **traversal algorithms**.

We will describe three of the most commonly used such algorithms, **preorder traversal, inorder traversal,** and **postorder traversal.**

Each of these algorithms can be defined recursively

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 $T_1, T_2, \ldots, T_n$ 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

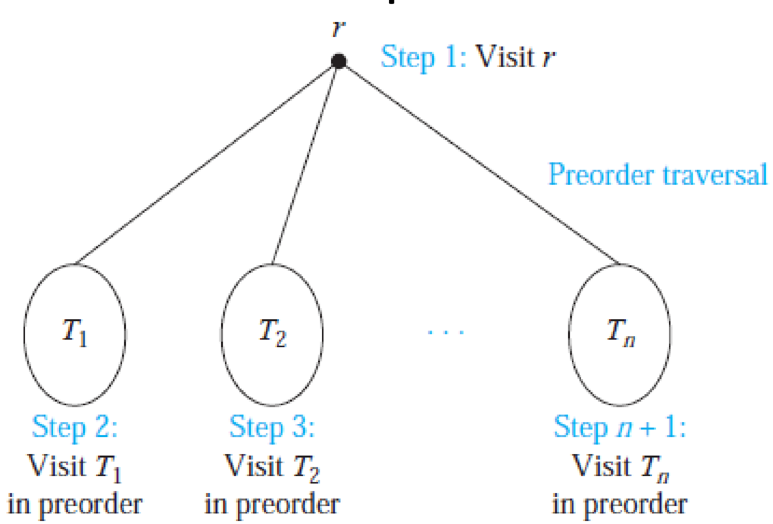


FIGURE 2 **Preorder Traversal.**

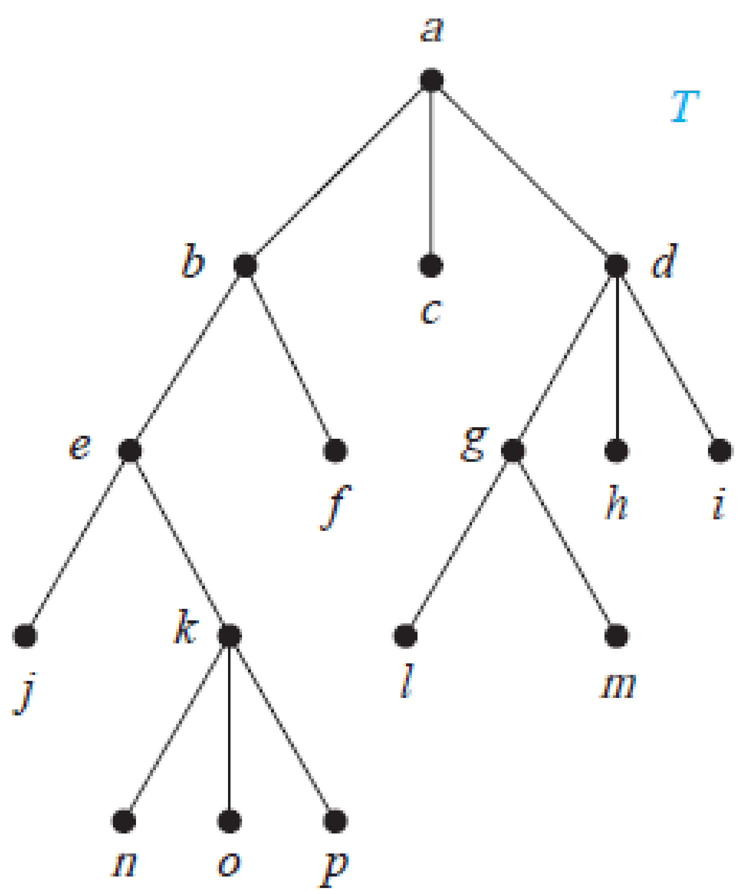


FIGURE 3 **The Ordered Rooted Tree *T*.**

Preorder Traversal

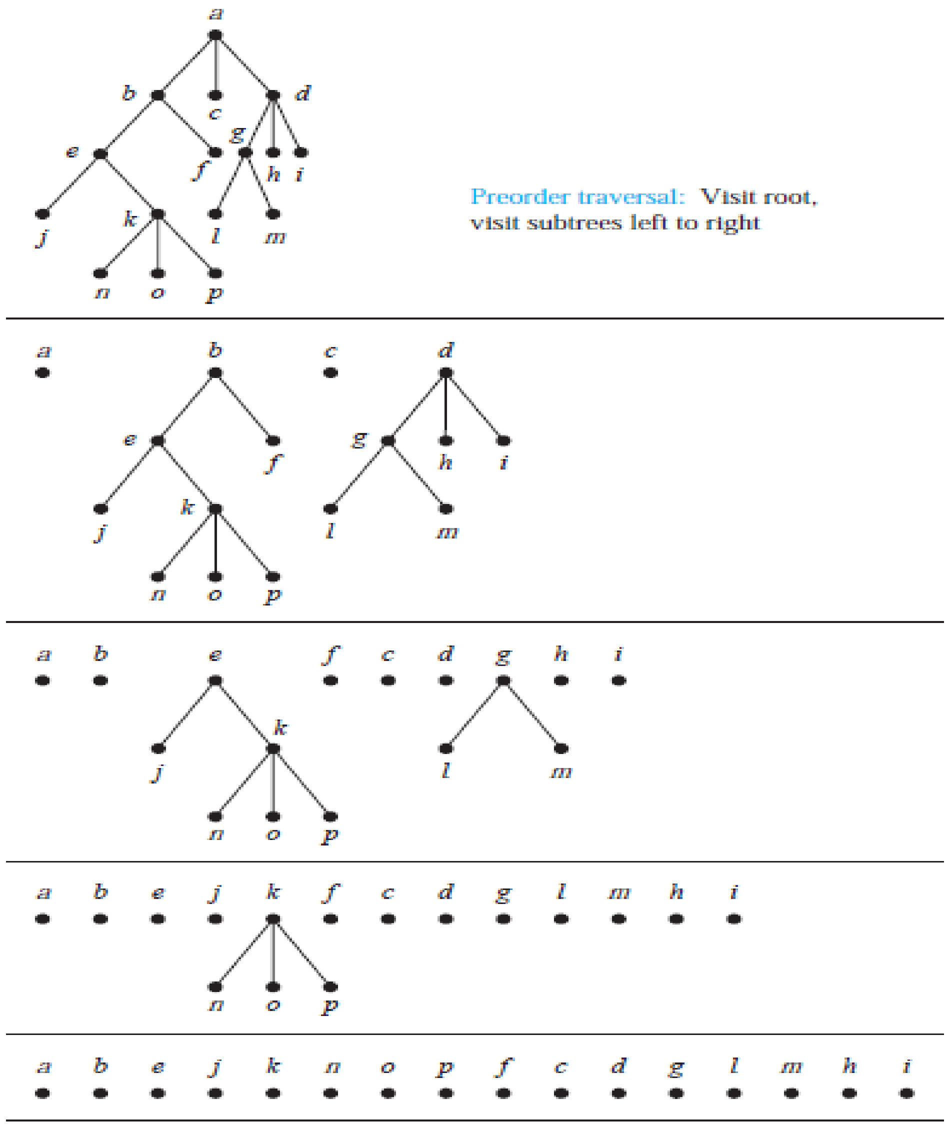


FIGURE 4 The Preorder Traversal of T.

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_n are the subtrees at r from **left to right**. The *inorder traversal* begins by traversing T_1 in inorder, then visiting r. It continues by traversing T_2 in inorder, then T_3 in inorder, . . . , and finally T_n in inorder.

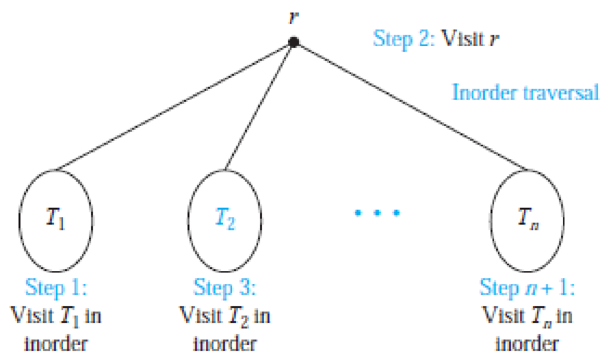


FIGURE 5 **Inorder Traversal.**

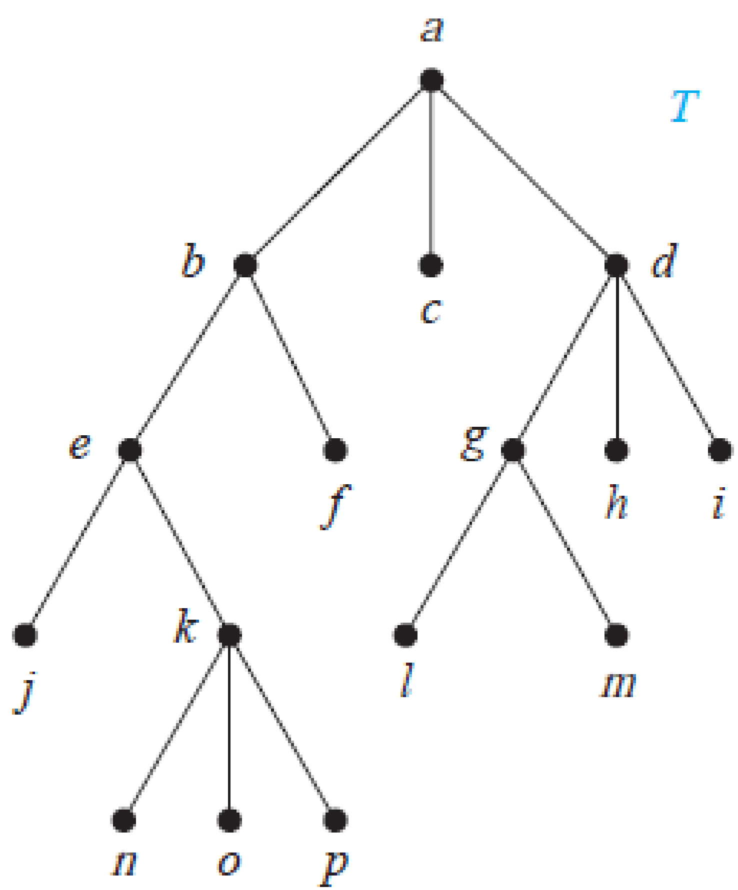


FIGURE 3 **The Ordered Rooted Tree *T*.**

Inorder Traversal

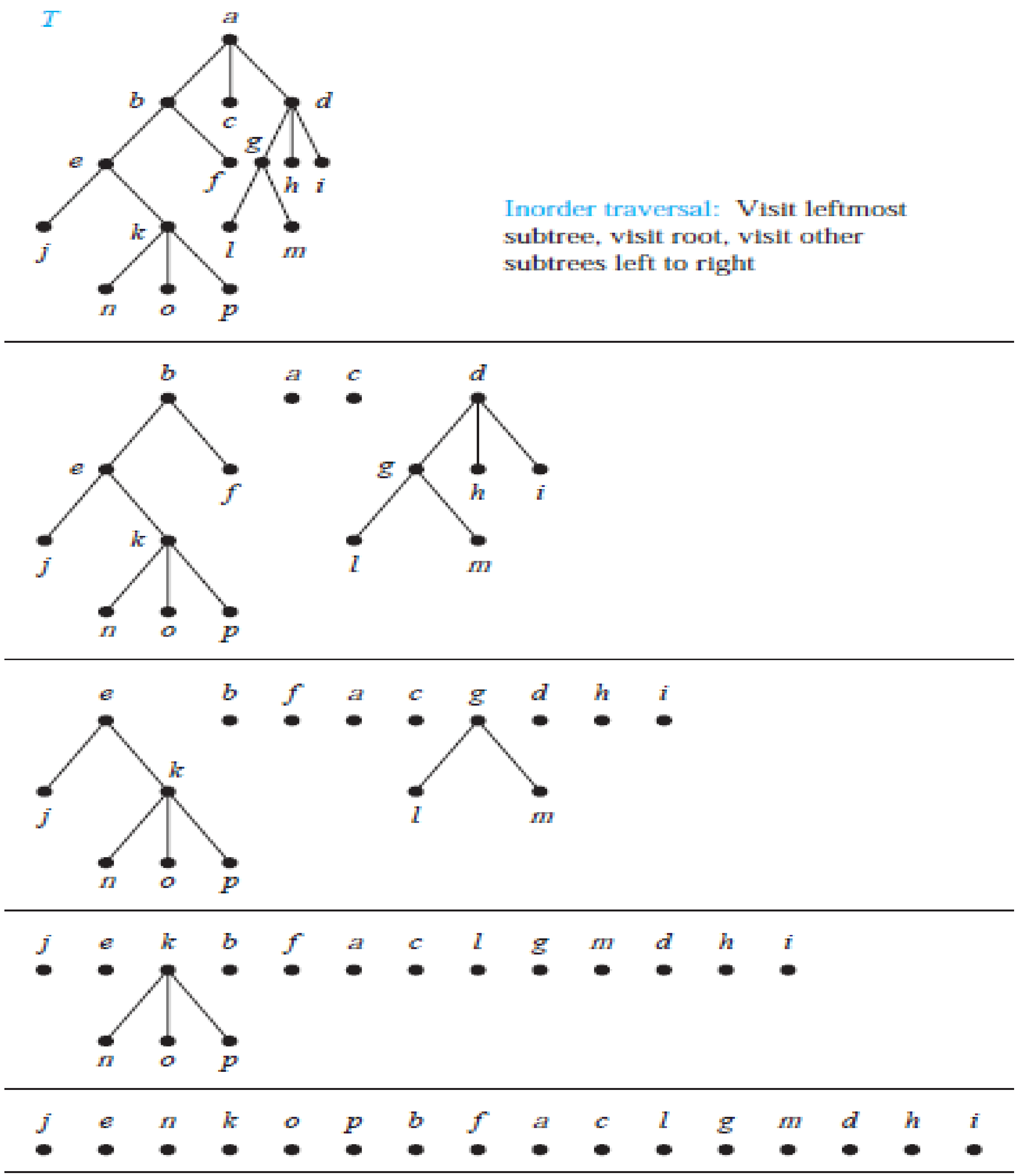


FIGURE 6 The Inorder Traversal of T.

Post order 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 $T_1, T_2, \ldots, T_n$ are the subtrees at r from left to right. The *postorder traversal* begins by traversing T_1 in postorder, then T_2 in postorder, . . . , then T_n in postorder, and ends by visiting r.

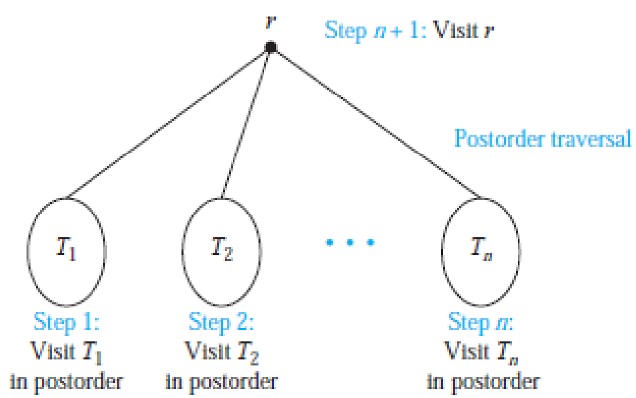


FIGURE 7 Postorder Traversal.

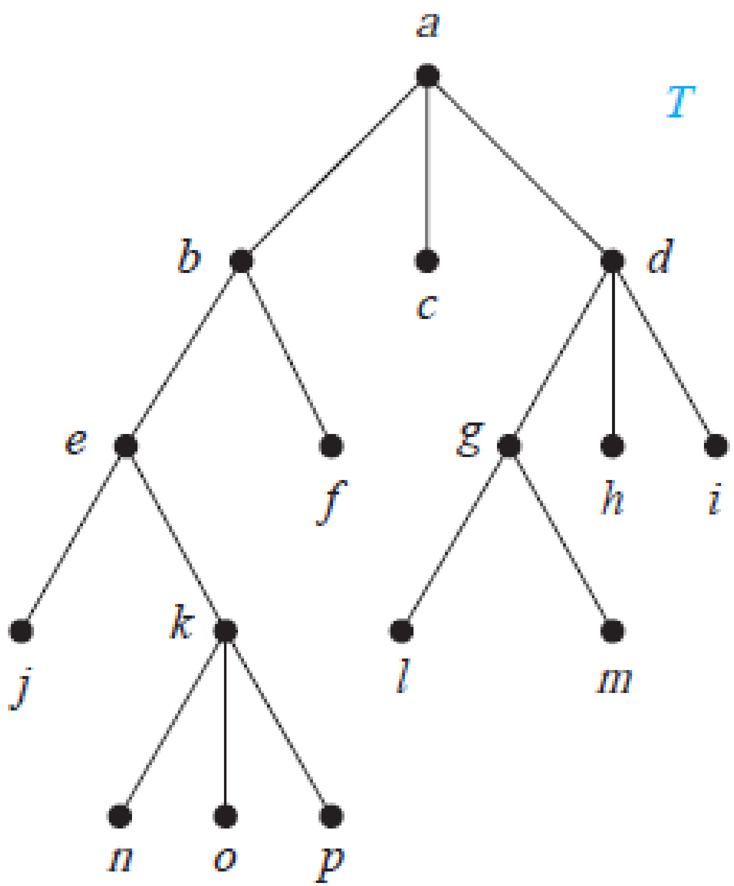


FIGURE 3 **The Ordered Rooted Tree *T*.**

Postorder

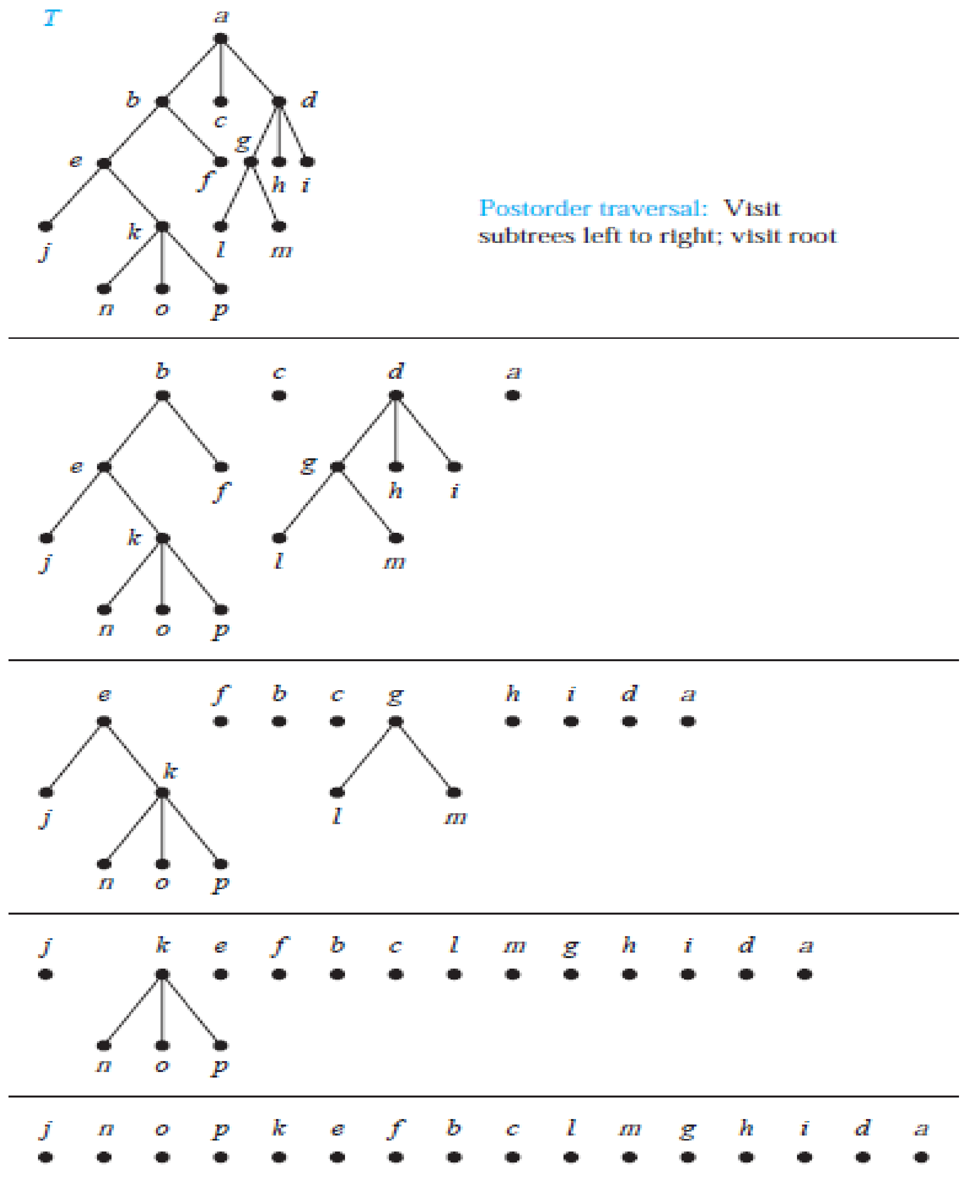


FIGURE 8 The Postorder Traversal of *T*.

Suggested Readings

10.2 Graph Terminology and Special Types of Graphs.