Introduction to Trees (Cont.)

Course Code: Course Title:

Dept. of Computer Science
Faculty of Science and Technology

Lecturer No:	**21**	**Week No:**	**12**	**Semester:**	
Lecturer:	*Md. Mahmudur Rahman (mahmudur@aiub.edu)*				

Lecture Outline

9.1 Introduction to Trees

Objectives and Outcomes

- Objectives: To introduce Tree and Forest and different jargons associated with that. To educate types of Trees and Forest to the student. Explain theorem related to tree. To demonstrate student how to solve mathematical problems by using different theorem and formula.
- Outcomes: After this class the student will be able to define different kind Tree and Forest. They will be capable of demonstrating theorems. Finally, by using Tree linked theorem they will be able to sort out math problems.

Example 2

- In the rooted tree T (with root a) shown in Figure 5, find the ***parent*** of c, the ***children*** of g, the ***siblings*** of h, all ***internal vertices***, and all ***leaves***. What is the ***subtree*** rooted at g?

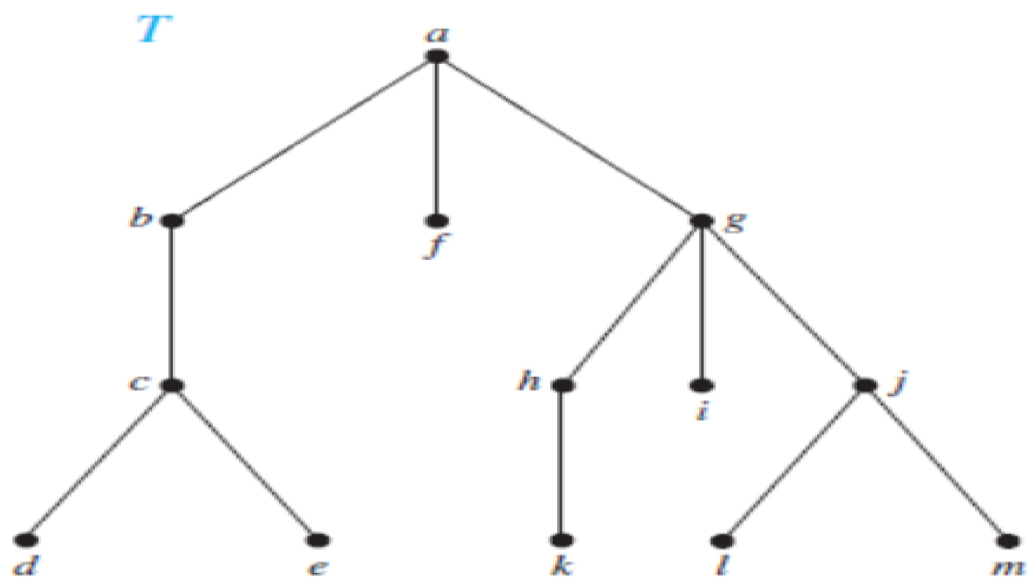


FIGURE 5 A Rooted Tree T.

Solution of EXAMPLE 2

- The **parent** of c is b.
- The **children** of g are h, i, and j .
- **The siblings** of h are i and j .
- **The internal vertices** are a, b, c, g, h, and j .
- The **leaves** are d, e, f , i, k, l, and m.
- The **subtree** rooted at g is shown in Figure 6(next slide)

Solution of EXAMPLE 2

The **subtree** rooted at g is shown in Figure 6

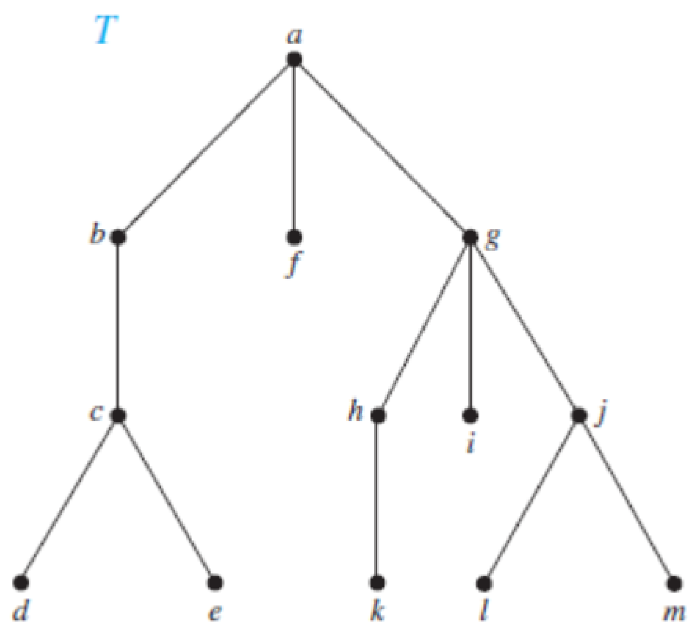


FIGURE 5 A Rooted Tree T.

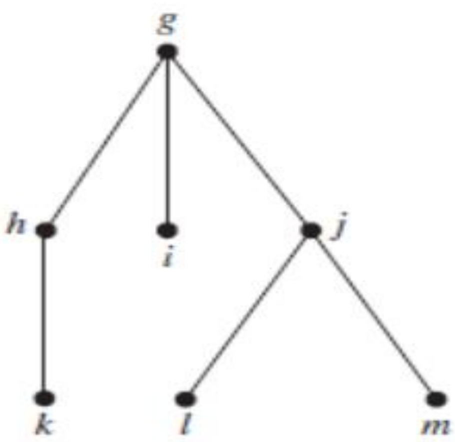


FIGURE 6 The Subtree Rooted at g.

m-ary tree, full m-ary tree, binary 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 3

- Are the rooted trees in Figure 7 **full *m-ary* trees** for some positive integer m?

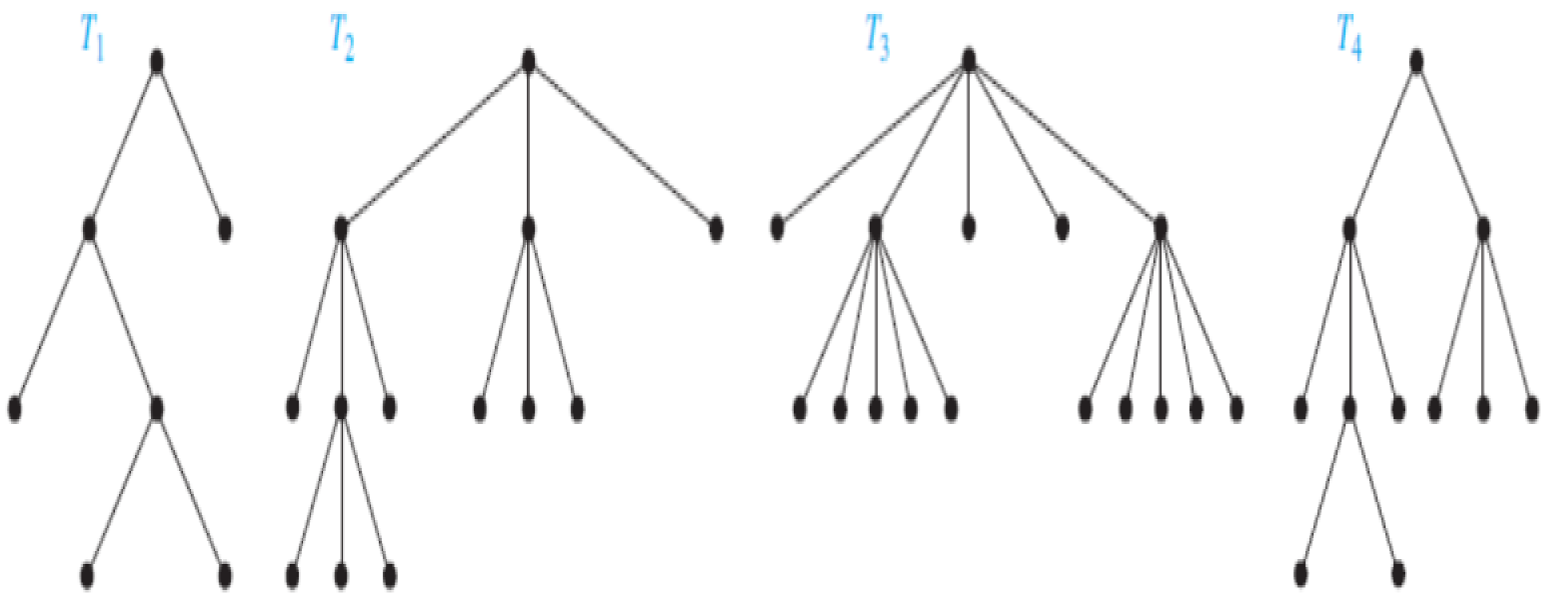


FIGURE 7 Four Rooted Trees.

Solution of Example 3

- T1 is a full binary tree because each of its internal vertices has two children.
- T2 is a full 3-ary tree because each of its internal vertices has three children.
- In T3 each internal vertex has five children, so T3 is a full 5-ary tree.
- 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.
- **Example 4 : Practice @ Home**

Properties of Trees

- **Theorem**: A tree with n vertices has $n - 1$ edges.

- **Theorem**: A full m-ary tree with i internal vertices contains $\boldsymbol{n = mi + 1}$ vertices.

Properties of Trees

Theorem: A full *m*-ary tree with

i. ***n* vertices** has $i = (n - 1)/m$ **internal vertices** and

$l = [(m - 1)n + 1] / m$ **leaves**,

ii. ***i* internal vertices** has $n = mi + 1$ **vertices** and

$l = (m - 1)i + 1$ **leaves**,

iii. ***l* leaves** has $n = (ml - 1)/(m - 1)$ **vertices** and

$i = (l - 1)/(m - 1)$ **internal vertices**

Exercise

17. How many **edges** does a tree with 10,000 vertices have?
18. How many **vertices** does a full 5-ary tree with 100 internal vertices have?
19. How many **edges** does a full binary tree with 1000 internal vertices have?
20. How many **leaves** does a full 3-ary tree with 100 vertices have?
27. Construct a complete binary tree of height 4 and a complete 3-ary tree of height 3.

Exercise

17. How many **edges** does a tree with 10,000 vertices have?

- **Solution**:

  Here, $n = 10000$

  $e = n - 1 = 9999$

Exercise

18. How many **vertices** does a full 5-ary tree with 100 internal vertices have?

- **Solution**:

  Here, $m = 5$, $i = 100$

  Therefore, $n = mi + 1 = 5.100 + 1 = 501$

Exercise

19. How many **edges** does a full binary tree with 1000 internal vertices have?

- **Solution**:

  Here, $m = 2$, $i = 1000$

  So, $n = mi + 1$

  $= 2.1000 + 1$

  $= 2001$

  Therefore, $e = n - 1$

  $= 2001 - 1$

  $= 2000$

Exercise

20. How many **leaves** does a full 3-ary tree with 100 vertices have?

- **Solution**:

  Here, $m = 3$, $n = 100$

  $$\mathbf{l = [(m-1)n + 1\,]/m}$$

  $$= [(3-1)100 + 1\,]/\,3$$

  $$= 201/3$$

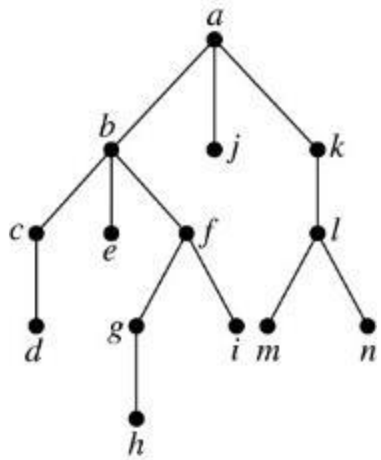
  $$= 67$$

Level of vertices and height of Trees

- When working with trees, we often want to have rooted trees where the subtrees at each vertex contain paths of approximately the same length.
- To make this idea precise we need some definitions:
  - The ***level*** of a **vertex *v*** in a rooted tree is the **length of the unique path from the root to this vertex**.
  - The ***height*** of a rooted tree is the **maximum of the levels of the vertices.**

Level of vertices and height of Trees

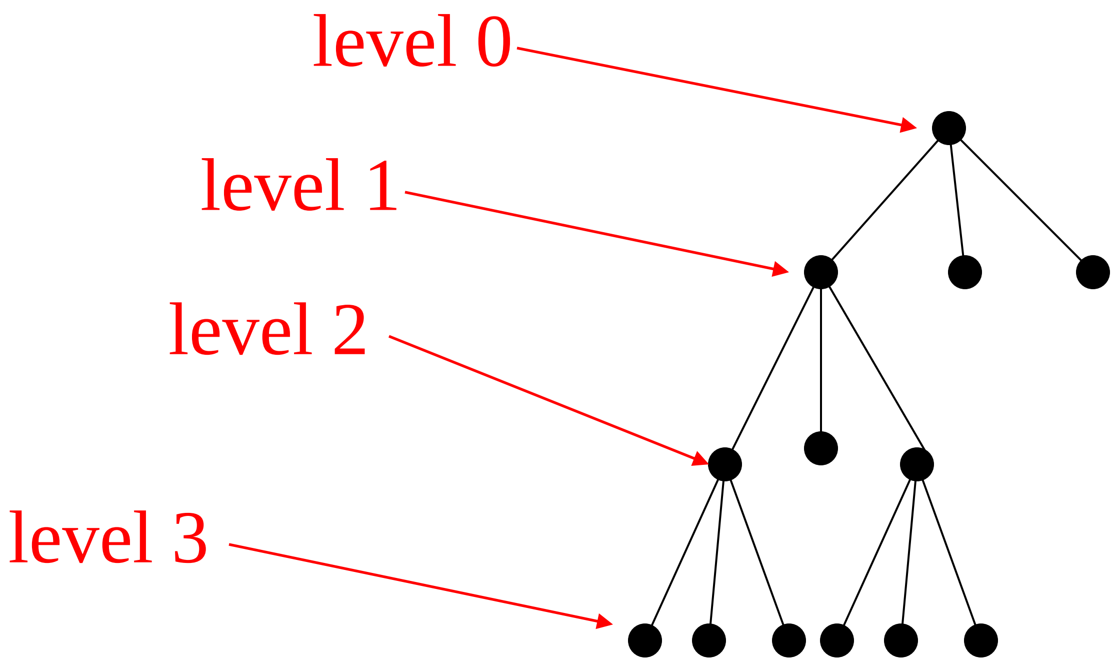
Example:
(*i*) Find the level of each vertex in the tree to the right.
(*ii*) What is the height of the tree?



Solution:

(*i*) The root *a* is at level 0. Vertices *b*, *j*, and *k* are at level 1. Vertices *c*, *e*, *f*, and *l* are at level 2. Vertices *d*, *g*, *i*, *m*, and *n* are at level 3. Vertex *h* is at level 4.
(*ii*) The height is 4, since 4 is the largest level of the vertices in this tree (vertex h).

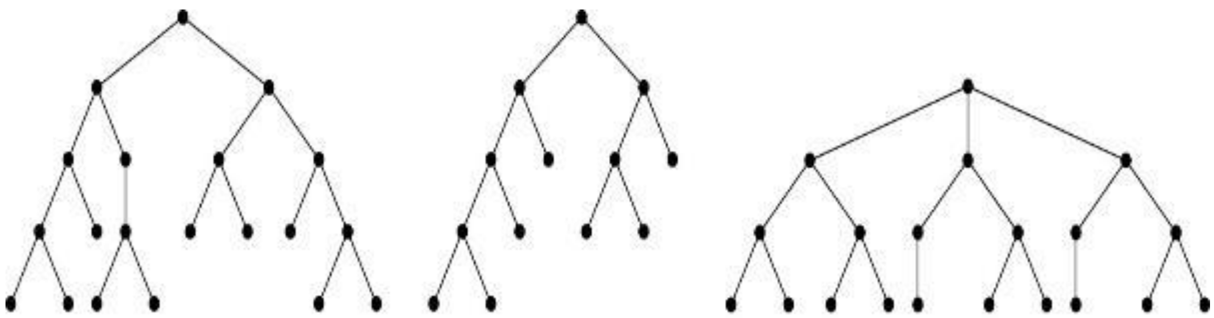
Illustration of Level of vertices



Balanced m-ary Trees

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

Example: Which of the rooted trees shown below is balanced?



T_1 T_2 T_3

Solution: T_1 and T_3 are balanced, but T_2 is not because it has leaves at levels 2, 3, and 4.

The Bound for the Number of Leaves in an m-ary Tree

- **Theorem 5**: There are **at most m^h leaves** in an m-ary tree of height h.

Books

- **Rosen, K. H., & Krithivasan, K. (2012). Discrete mathematics and its applications: with combinatorics and graph theory. Tata McGraw-Hill Education. (7th Edition)**
- Liu, C. L. (1986). Elements of discrete mathematics. Tata McGraw-Hill Education.

References

1. Discrete Mathematics, *Richard Johnsonbaugh*, Pearson education, Inc.
2. Discrete Mathematical Structures, *Bernard Kolman*, *Robert C. Busby*, *Sharon Ross,* Prentice-Hall, Inc.
3. *SCHAUM'S outlines Discrete Mathematics(2nd edition)*, by *Seymour Lipschutz*, *Marc Lipson*

- University of Wisconsin-Madison

http://pages.cs.wisc.edu/~deppeler/cs367-common/readings/Trees/intro.html

- Bradfield School of Computer Science

https://bradfieldcs.com/algos/trees/introduction/