Heaven's light is our guide"

Rajshahi University of Engineering & Technology
Department of Computer Science & Engineering

Discrete Mathematics
Course No. : 305
Chapter 9: Trees
Prepared By : Julia Rahman

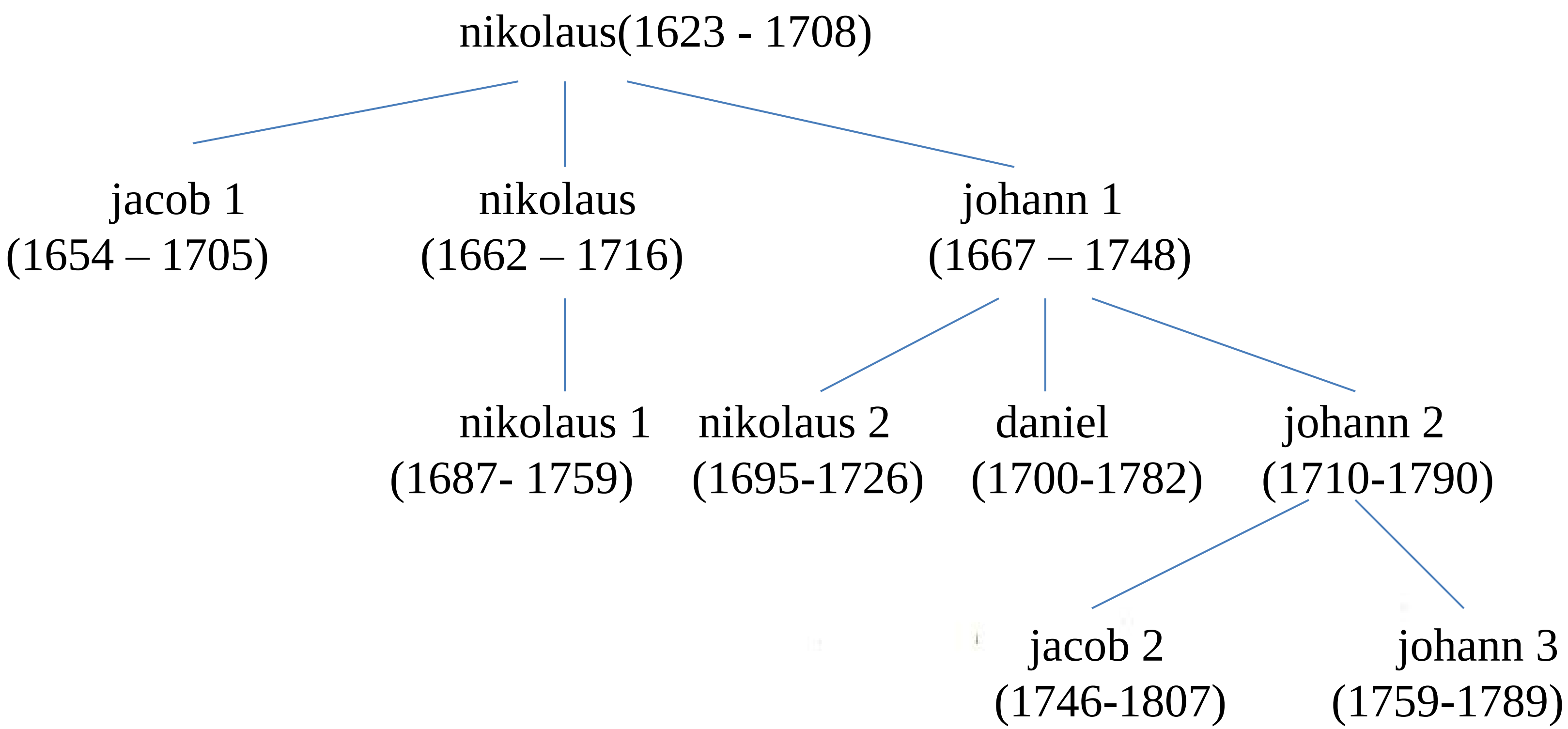
9.1 Introduction to Trees

- **Tree:**
  - ✓ A *tree* is a connected undirected graph with no simple circuits.
  - ✓ A tree cannot have a simple circuit.
  - ✓ A tree cannot contain multiple edges or loops.

Recall:

A **circuit** is a path of length >=1 that begins and ends a the same vertex.

Example of tree:

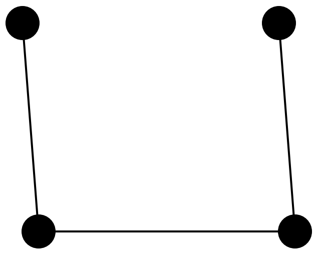


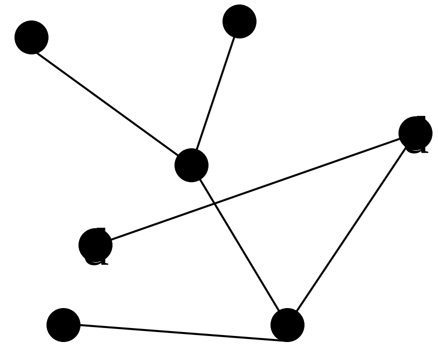
9.1 Introduction to Trees

- **Theorem:**

  An undirected graph is a tree if and only if there is a unique simple path between any two of its vertices.

  Like:



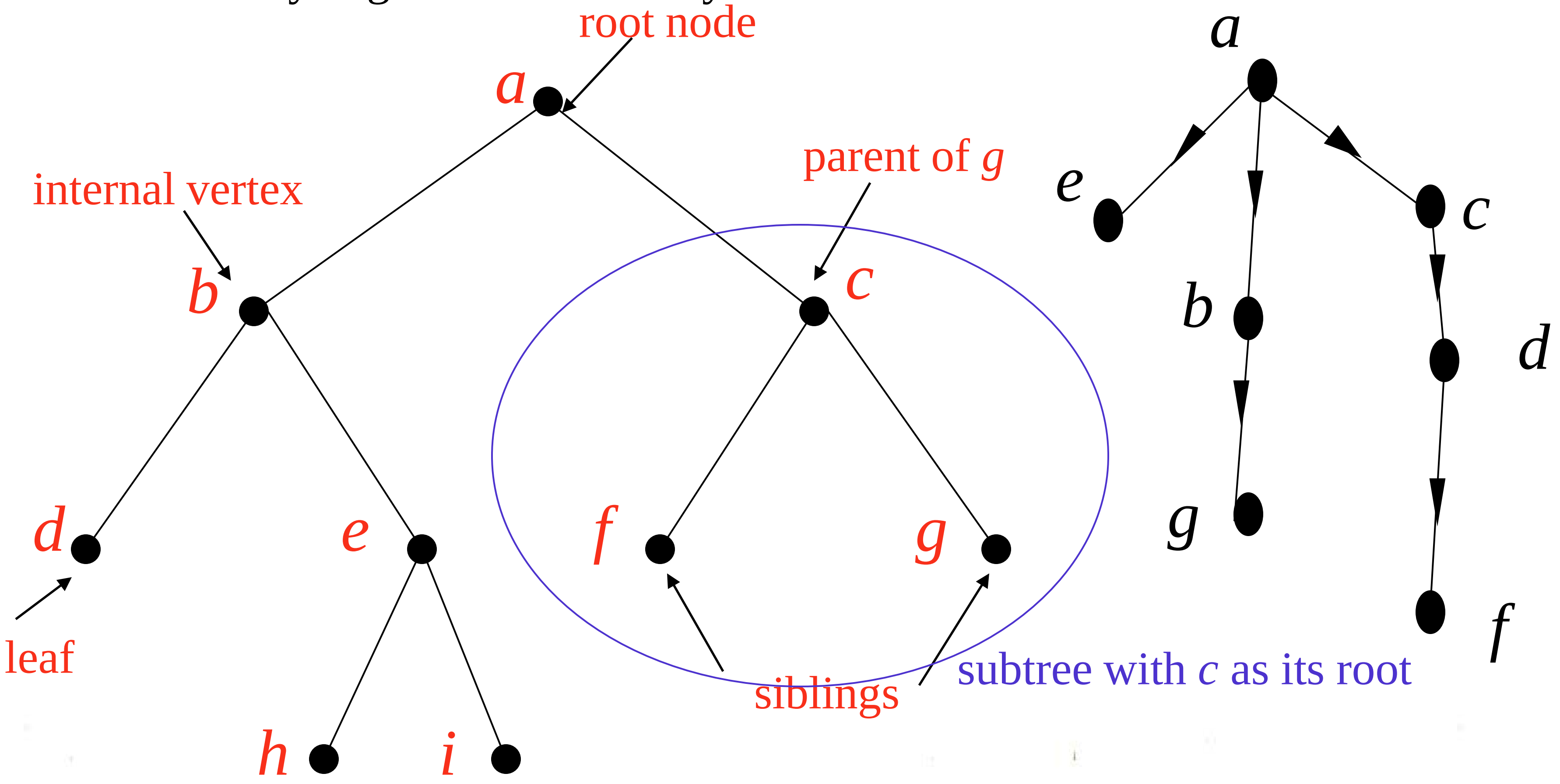


- **DEFINITION 1(Forest):**

  Graphs containing no simple circuits that are not connected, but each of their connected component is a tree.

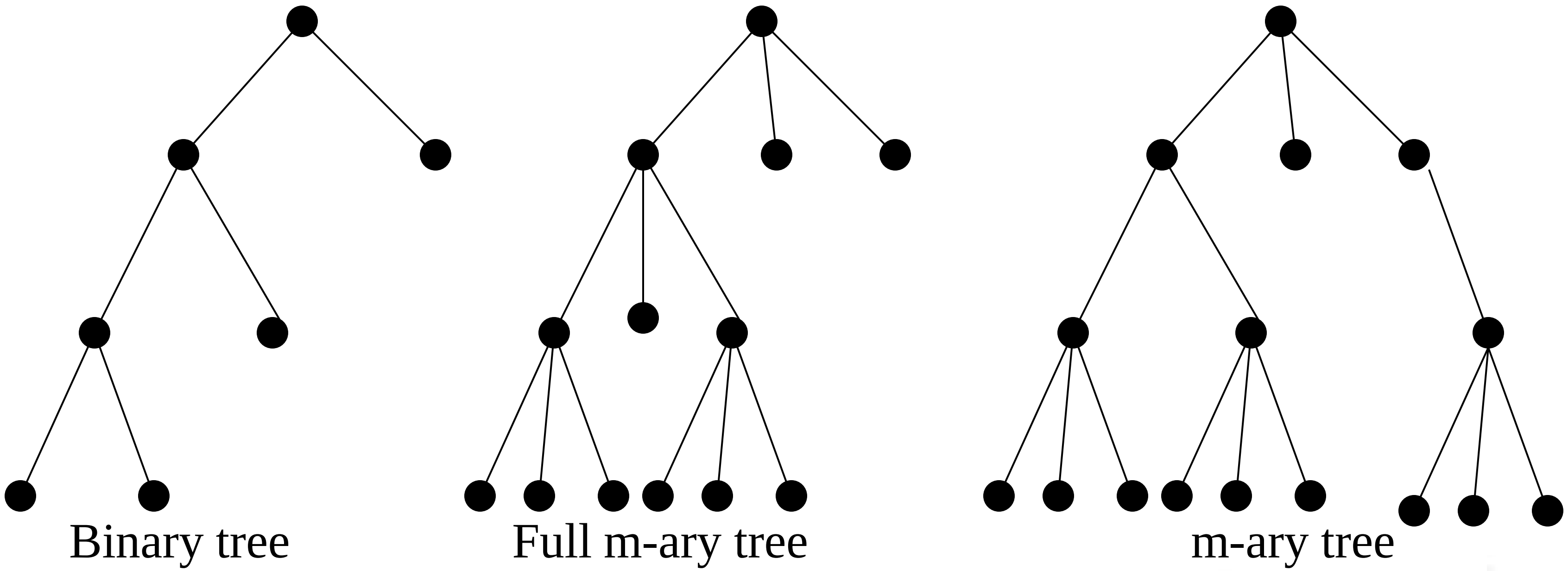
9.1 Introduction to Trees

- **Rooted Trees:**
  - ✓ Once a vertex of a tree has been designated as the *root* of the tree and every edge is directed away from root.



9.1 Introduction to Trees

- ***m*-ary trees:**
  - ✓ 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***.



9.1 Introduction to Trees

- **Ordered Rooted Tree:**
  - ✓ An ordered rooted tree is a rooted tree where the children of each internal vertex are ordered.
  - ✓ Ordered trees are drawn so that the children of each internal vertex are shown in order from left to right.
- **Trees as models:**
  1) Structured Hydrocarbons and trees
  2) Representing Organizations
  3) Computer file systems
  4) Tree connected parallel processors
- **Properties of Trees:**

  a) A tree with n vertices has n-1 edges.

  Here number of vertices is 11 and number of edges is 10.

  It follows the rules.

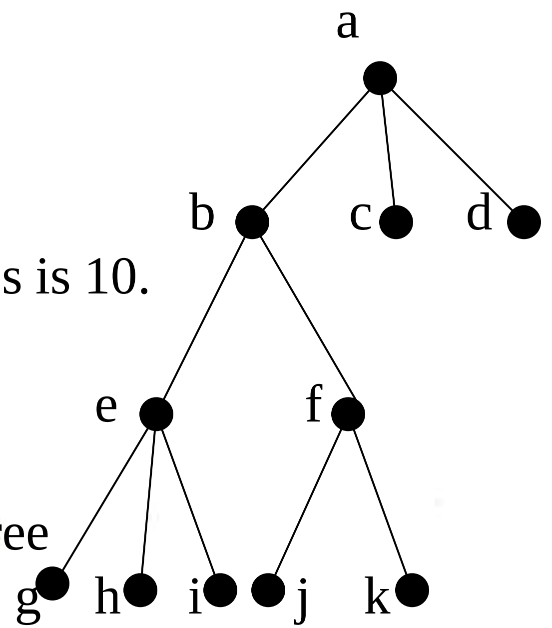


fig: Ordered tree

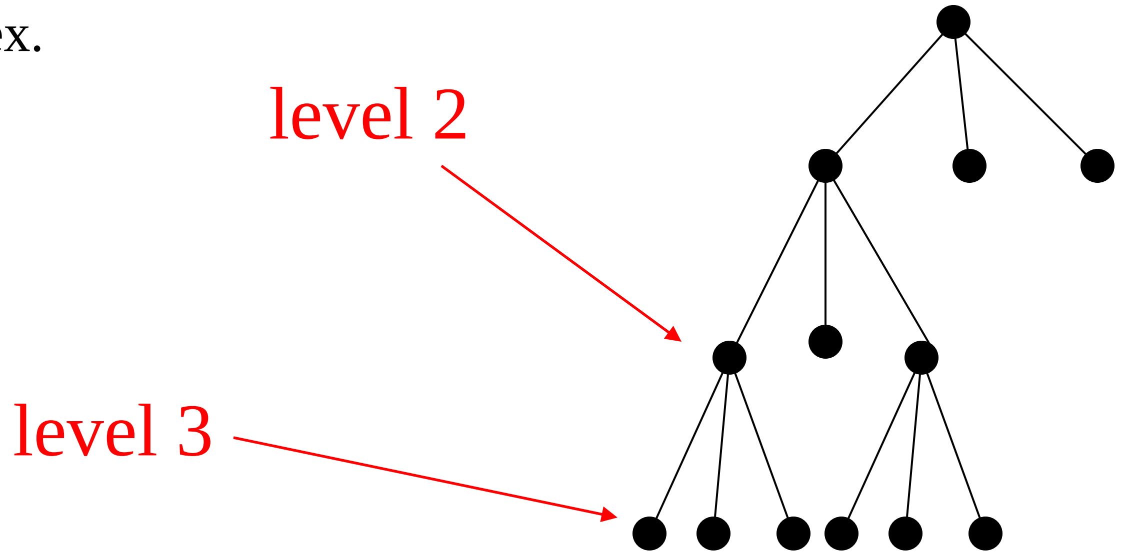
9.1 Introduction to Trees

b) A full m-ary tree with i internal vertices contains $n = mi+1$ vertices.
Here without root there are i internal vertices and each internal vertices has m children, so total vertices = $mi+1$.

c) 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.
  Proof: We know n = mi+1 (previous theorem) and n = l + i,
    n – no. vertices i – no. internal vertices l – no. leaves
    For example, i = (n-1)/m
    l = n – i = n – (n - 1)/m = [(m – 1)n + 1]/m

9.1 Introduction to Trees

d) The level of a vertex *v* in a rooted tree is the length of the unique path from the root to this vertex.



e) The *height* of a rooted tree is the maximum of the levels of vertices.

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

g) There are at most m^h leaves in an *m*-ary tree of height *h*.

h) If an *m*-ary tree of height *h* has *l* leaves, then

$$h \geq \lceil \log_m l \rceil$$

9.2 Applications of Trees

- **Binary Search Trees** to store items for easy retrieval, insertion and deletion. For balanced trees, each of these steps takes *log(N)* time for an *N* node tree. T is called binary search tree if each node N of T has the following property:

  The value of N is greater than every value in the left subtree of N and is less than every value in the left subtree.

- **Huffman trees** are used to compress data. They are most commonly used to compress faxes before transmission.

- **Spanning trees** are subgraphs that have applications to computer and telephone networks. **Minimum spanning trees** are of special interest.