Review of Data Structures

Virendra Singh

Professor, Indian Institute of Technology Bombay

And

Adjunct Professor, Indian Institute of Technology Jammu

http://www.ee.iitb.ac.in/~viren/

E-mail: viren@ee.iitb.ac.in, virendra.singh@iitjammu.ac.in

CSPL201: Data Organization & Retrieval

Lecture 6 (29 September 2020)

Acknowledgement

- Prof. Sartaj Sahni, Uni. of Florida
- Prof. Hideo Fujiwara, NAIST
- Late Prof. A. Bhattacharya, SERC, IISc

Data Structures

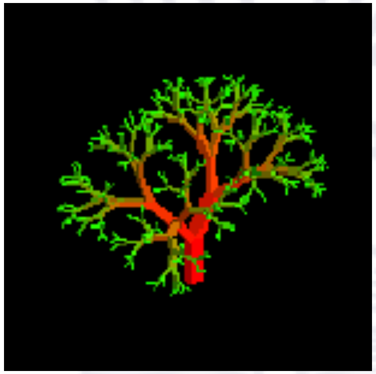
- Data structure is a way to store and organize data in order to facilitate access and modification
- No single data structure works well for all purposes

- data object
- set or collection of instances
  - integer = {0, +1, -1, +2, -2, +3, -3, …}
  - daysOfWeek = {S,M,T,W,Th,F,Sa}
- instances may or may not be related
  - myDataObject = {apple, chair, 2, 5.2, red, green, Jack}

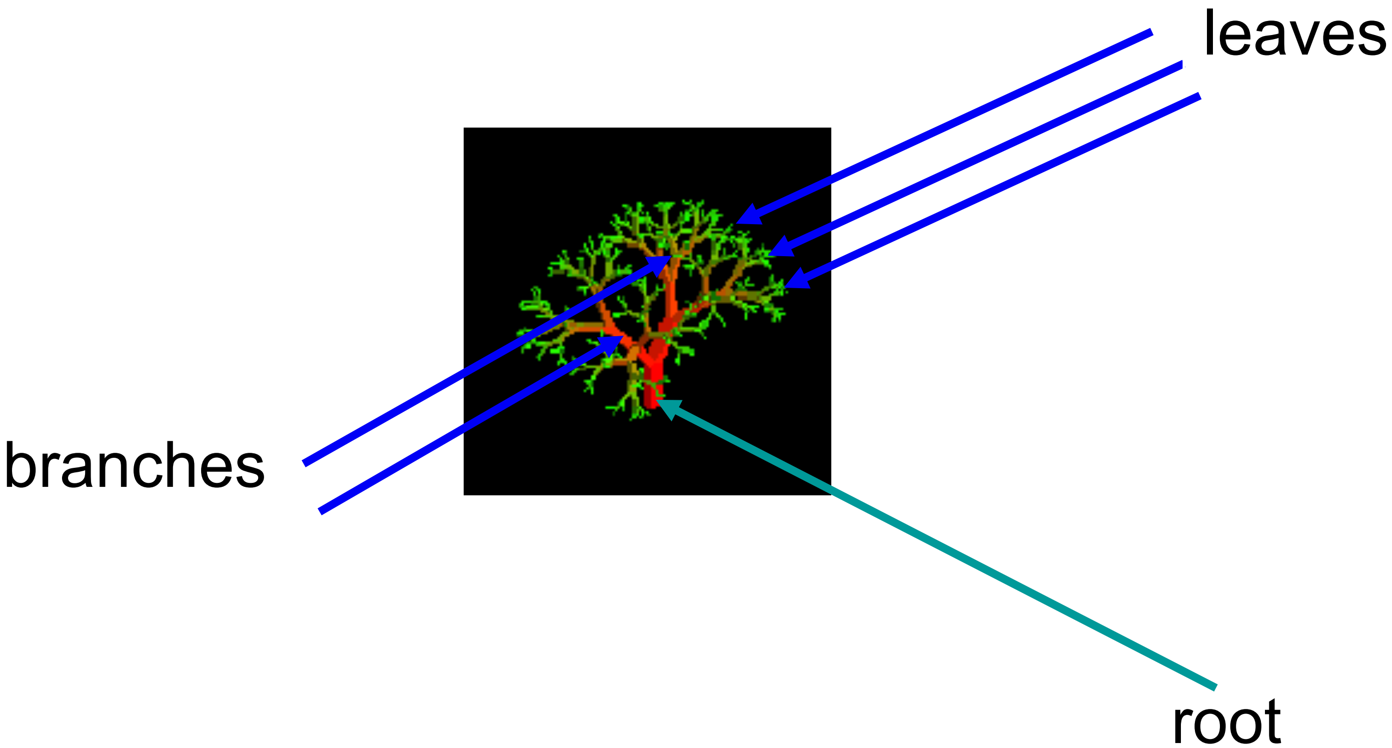
Dictionaries

- Collection of pairs.
  - (key, element)
  - Pairs have different keys.
- Operations.
  - get (theKey)
  - put (theKey, theElement)
  - remove (theKey)

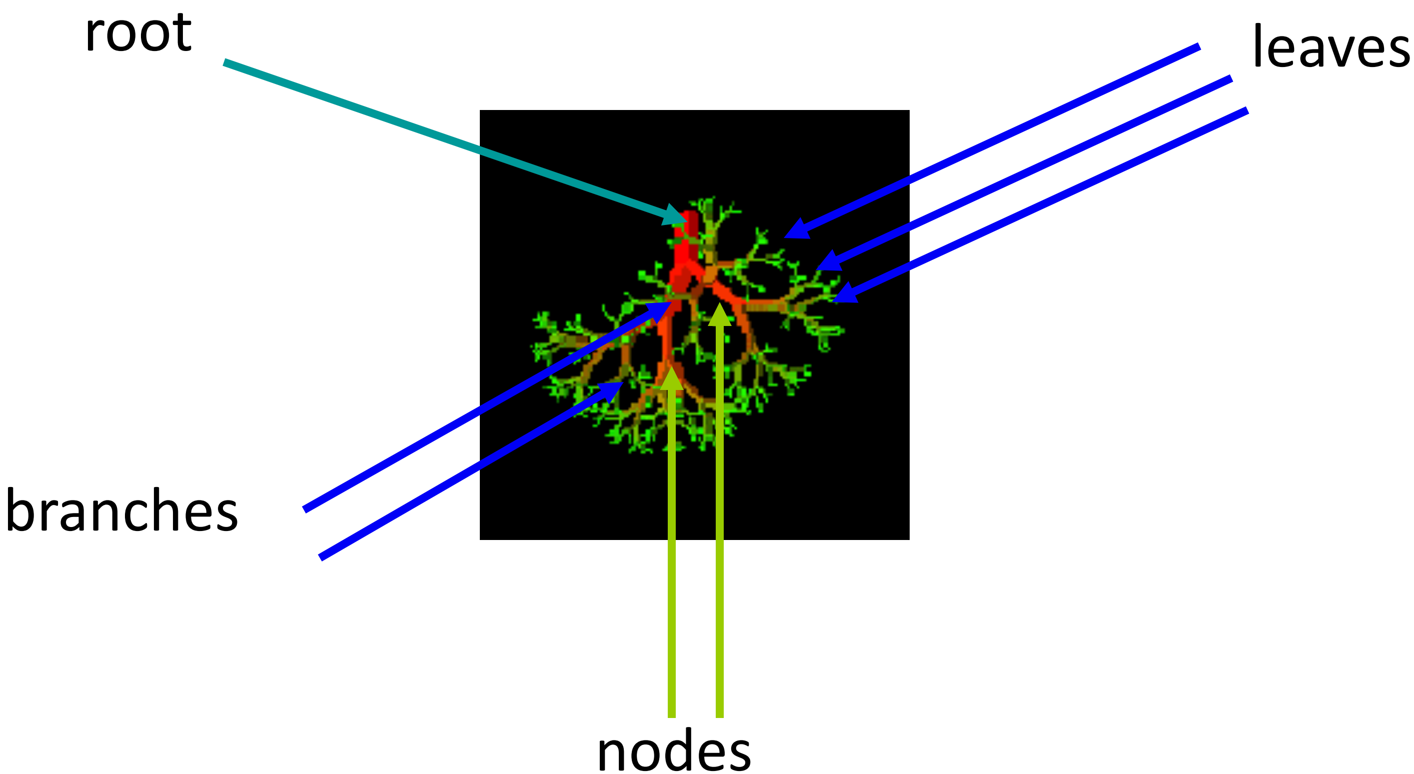
Trees



Nature Lover’s View of A Tree



Computer Scientist’s View



Linear Lists And Trees

- Linear lists are useful for serially ordered data.
  - $(e_0, e_1, e_2, \ldots, e_{n-1})$
  - Days of week.
  - Months in a year.
  - Students in this class.
- Trees are useful for hierarchically ordered data.
  - Employees of a corporation.
    - President, vice presidents, managers, and so on.
  - Java's classes.
    - Object is at the top of the hierarchy.
    - Subclasses of Object are next, and so on.

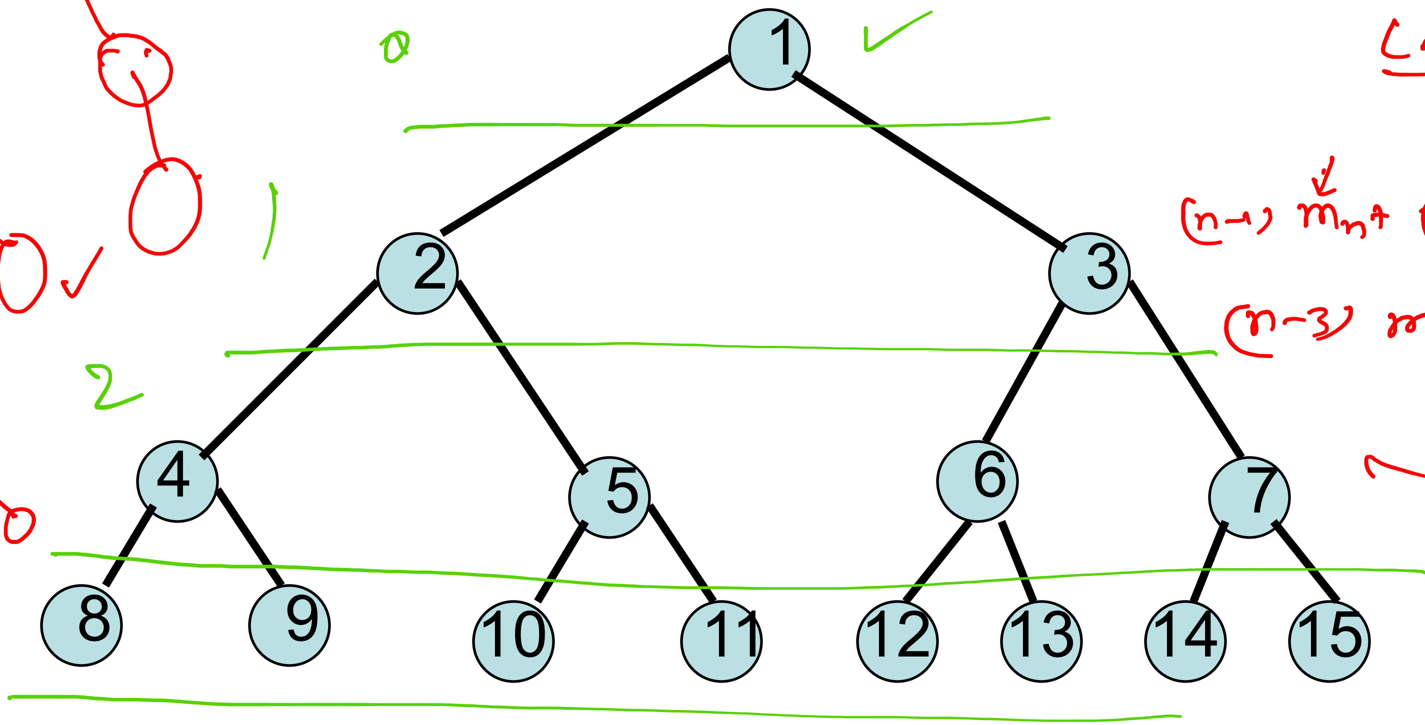
Hierarchical Data And Trees

- The element at the top of the hierarchy is the root.
- Elements next in the hierarchy are the children of the root.
- Elements next in the hierarchy are the and children of the root, and so on.
- Elements that have no children are .

Definition

- A tree is a finite nonempty set of elements.
- One of these elements is called the root.
- The remaining elements, if any, are partitioned into trees, which are called the subtrees of t.

Node Number Properties



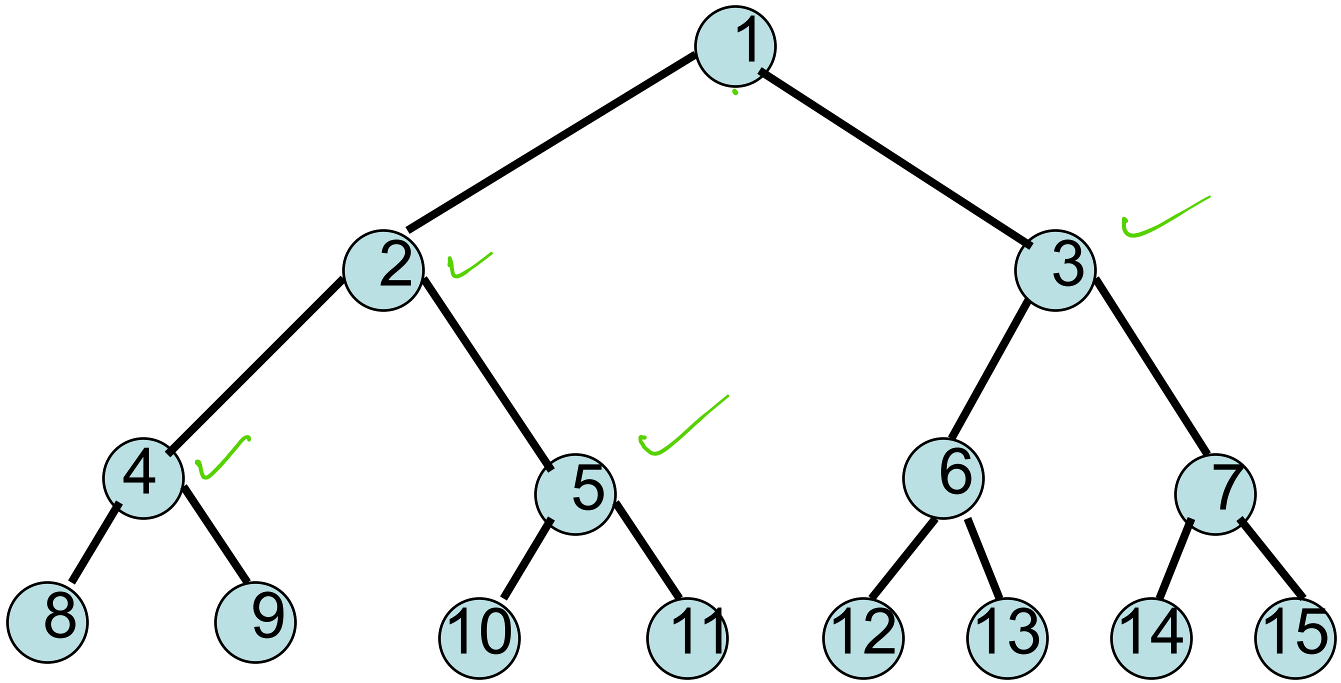
Leaf

$m_2 + 1$

$(n-1)\, m_n + (n-2)\, m_{n-1} + (n-3)\, m_{n-2} + \cdots + 2m_3 + m_2 + 1$

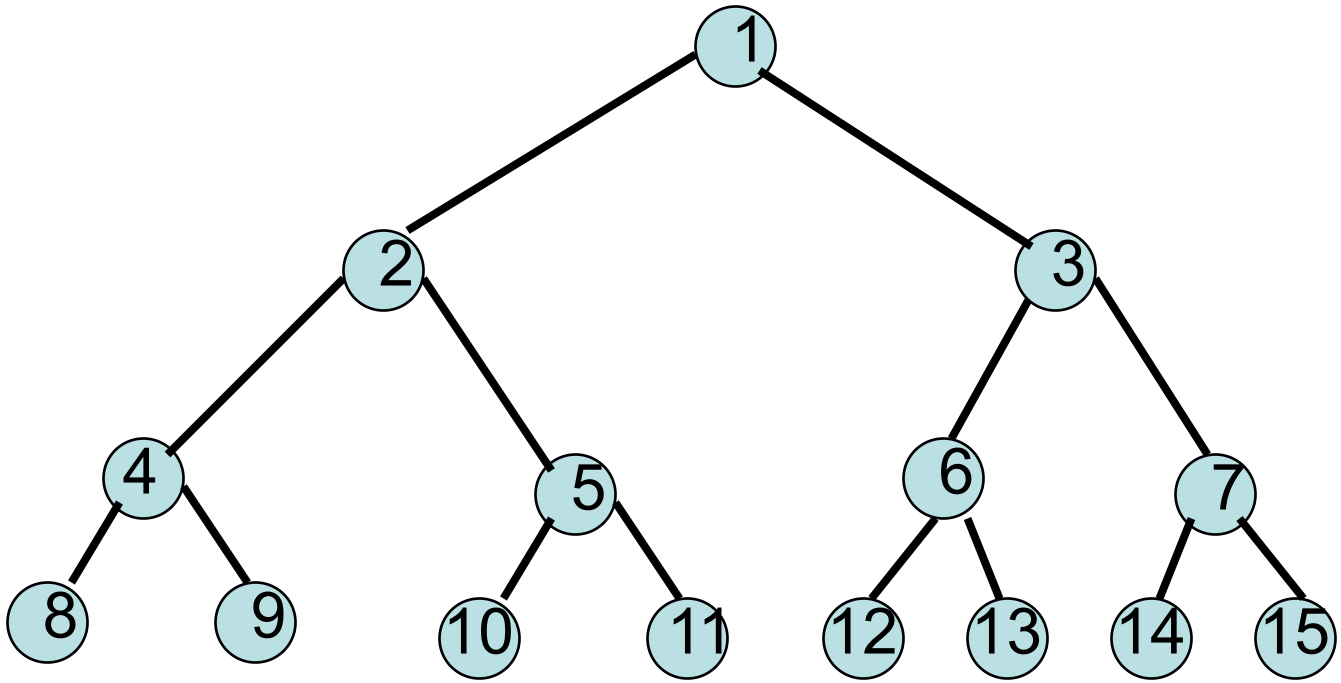
- Parent of node i is node i / 2, unless i = 1.
- Node 1 is the root and has no parent.

Node Number Properties



- Left child of node i is node $2i$, unless $2i > n$, where n is the number of nodes.
- If $2i > n$, node i has no left child.

Node Number Properties

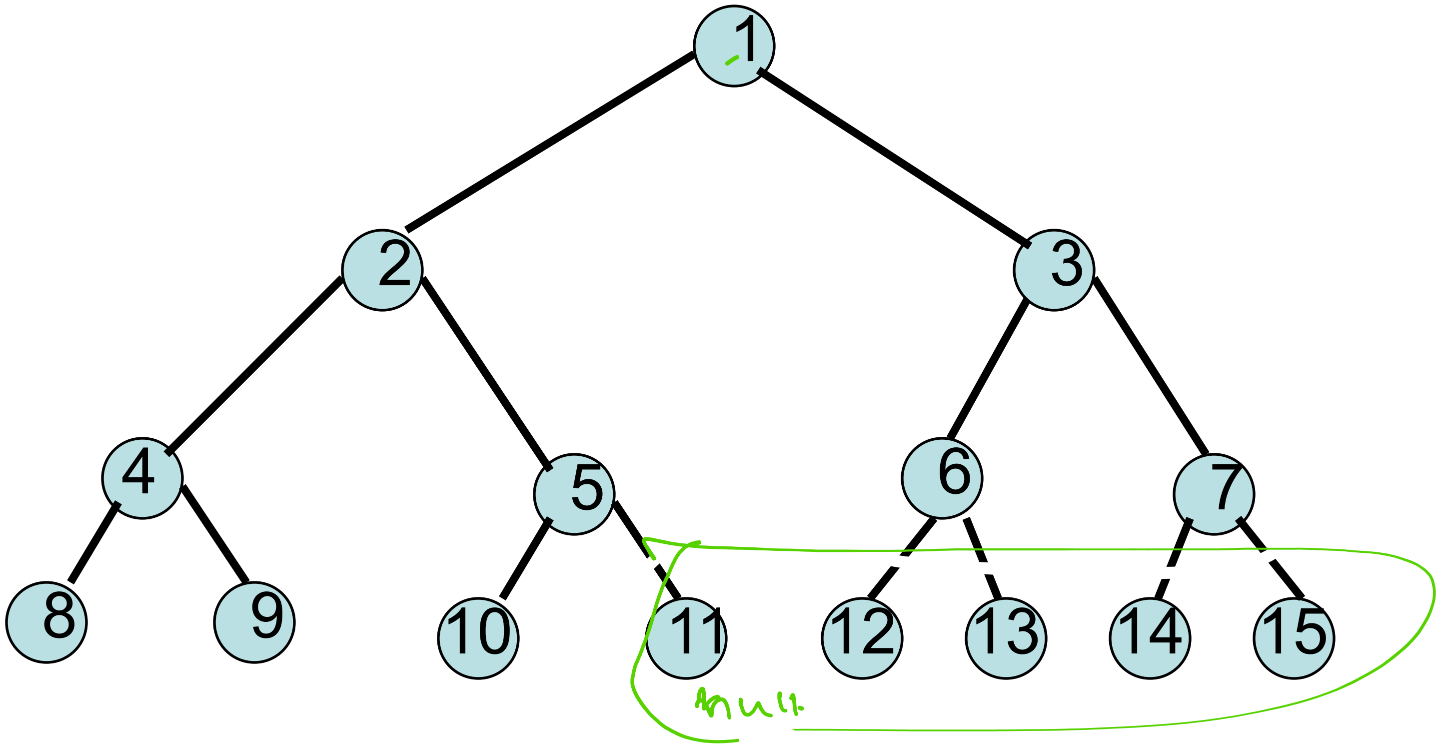


- Right child of node i is node $2i+1$, unless $2i+1 > n$, where n is the number of nodes.
- If $2i+1 > n$, node i has no right child.

Complete Binary Tree With n Nodes

- Start with a full binary tree that has at least n nodes.
- Number the nodes as described earlier.
- The binary tree defined by the nodes numbered 1 through n is the unique n node complete binary tree.

Example



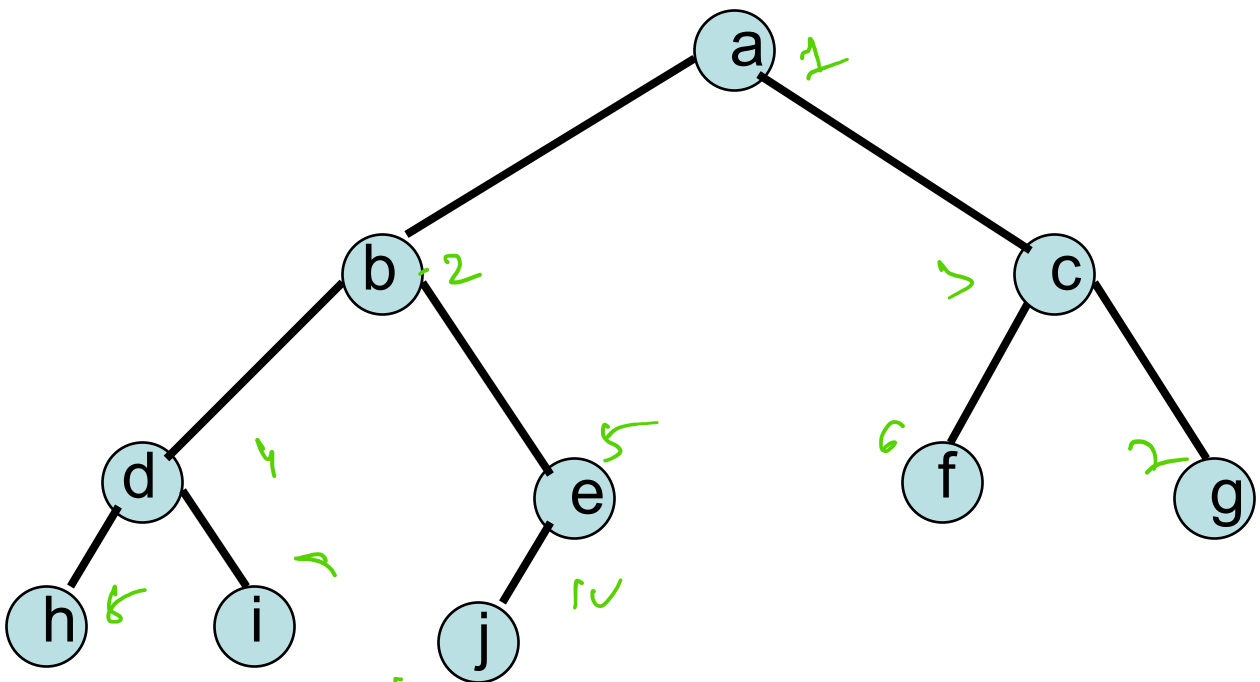
- Complete binary tree with 10 nodes.

Binary Tree Representation

- Array representation.
- Linked representation.

Array Representation

- Number the nodes using the numbering scheme for a full binary tree. The node that is numbered i is stored in tree[i].



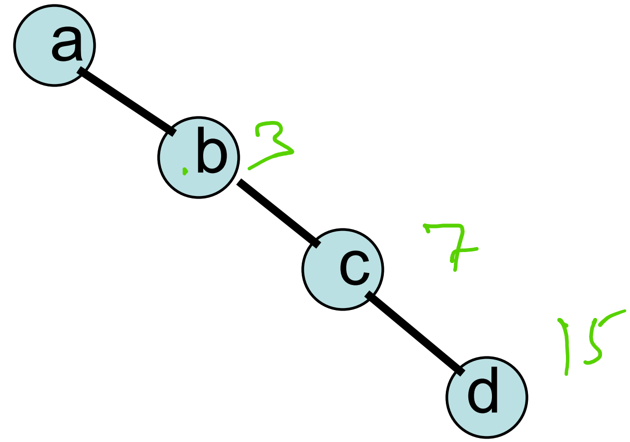
tree[]

	a	b	c	d	e	f	g	h	i	j

0

Compact

Right-Skewed Binary Tree



1

tree[]																
		a	-	b	-	-	-	c	-	-	-	-	-	-	-	d

✓

utilization of space
↓
POOR

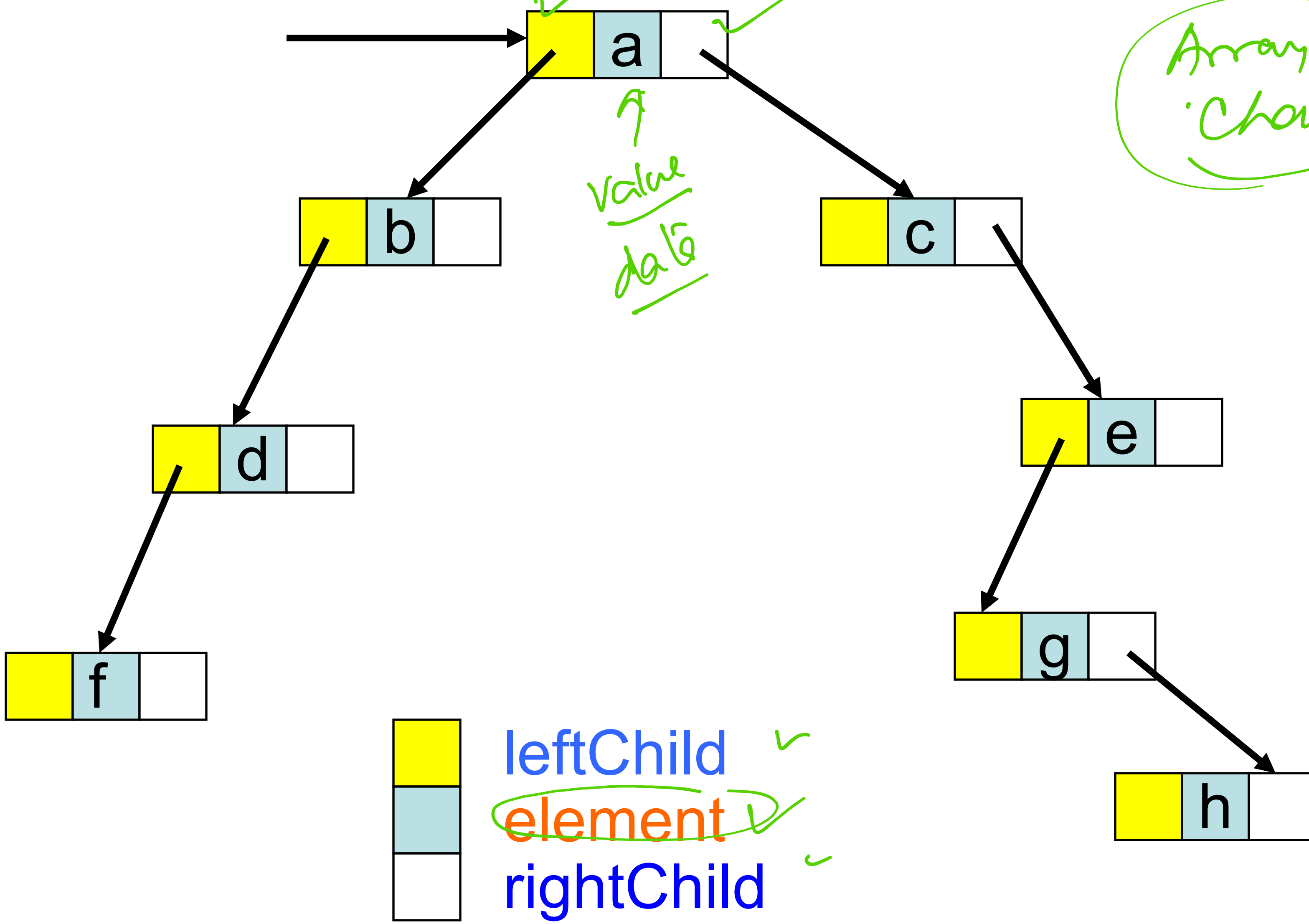
- An n node binary tree needs an array whose length is between $n+1$ and 2^n.

Linked Representation

Chain

- Each binary tree node is represented as an object whose data type is BinaryTreeNode.
- The space required by an n node binary tree is n * (space required by one node).

Linked Representation Example



Array 'Chain'

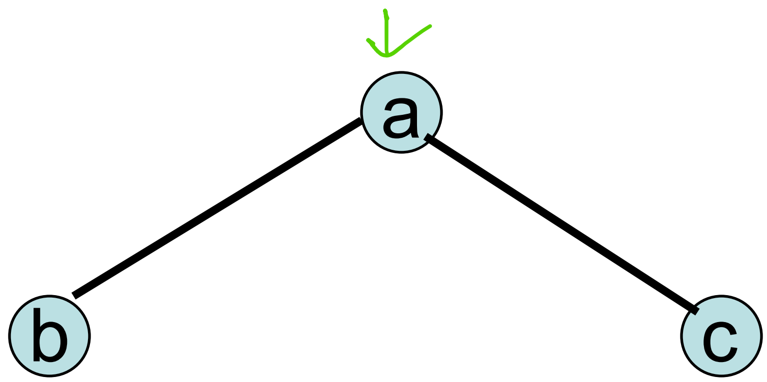
Binary Tree Traversal

- Many binary tree operations are done by performing a traversal of the binary tree.
- In a traversal, each element of the binary tree is visited exactly once.
- During the visit of an element, all action (make a clone, display, evaluate the operator, etc.) with respect to this element is taken.

Binary Tree Traversal Methods

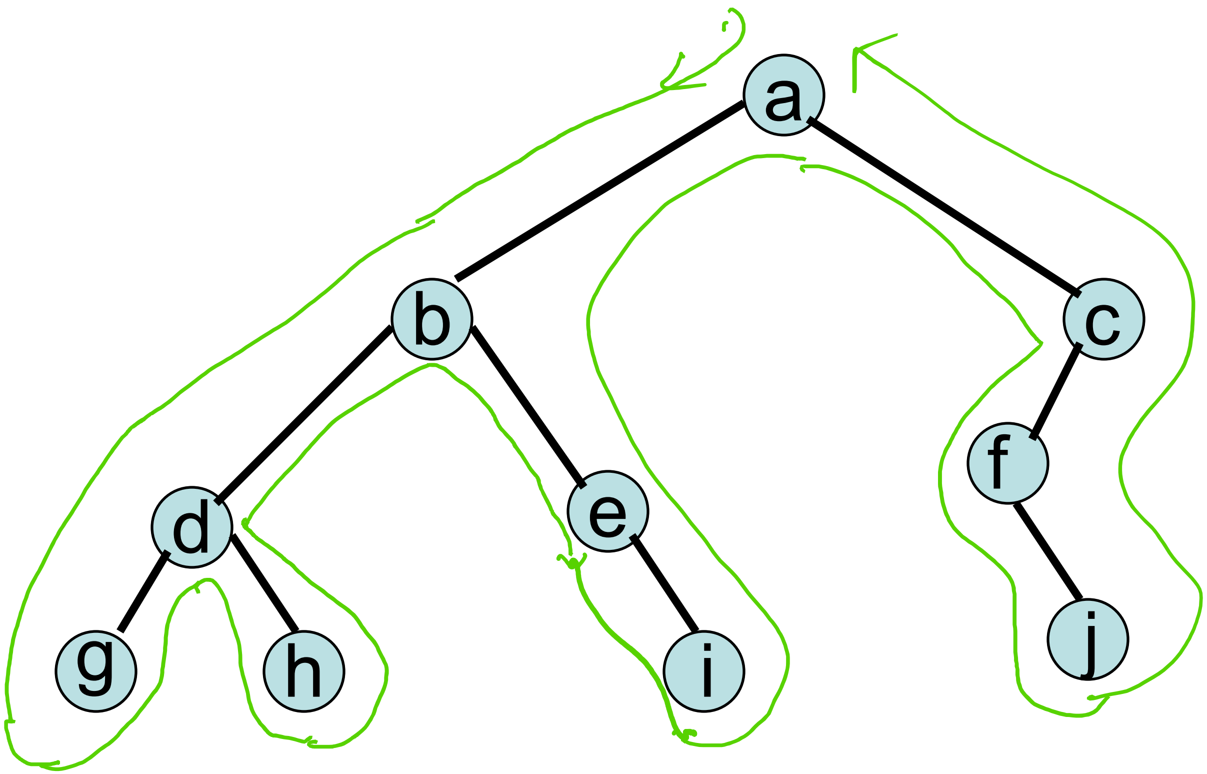
- Preorder
- Inorder
- Postorder
- Level order

Preorder Example (visit = print)



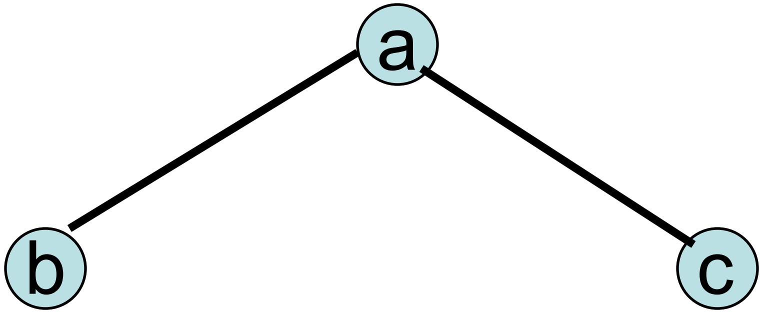
a b c

Preorder Example (visit = print)



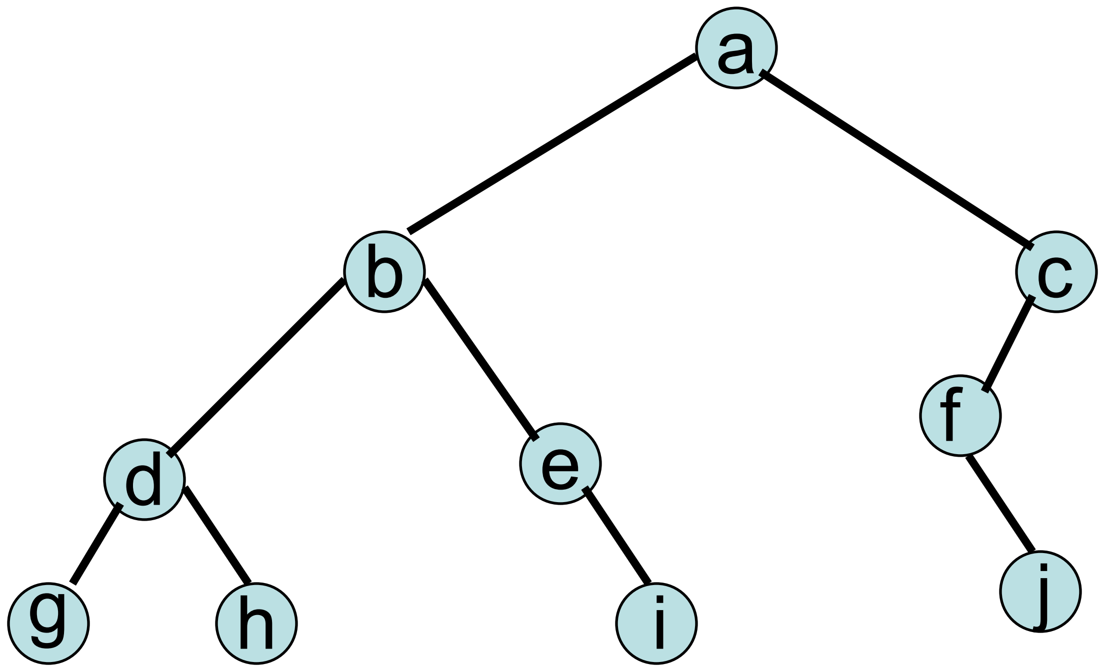
a b d g h e i c f j

Inorder Example (visit = print)



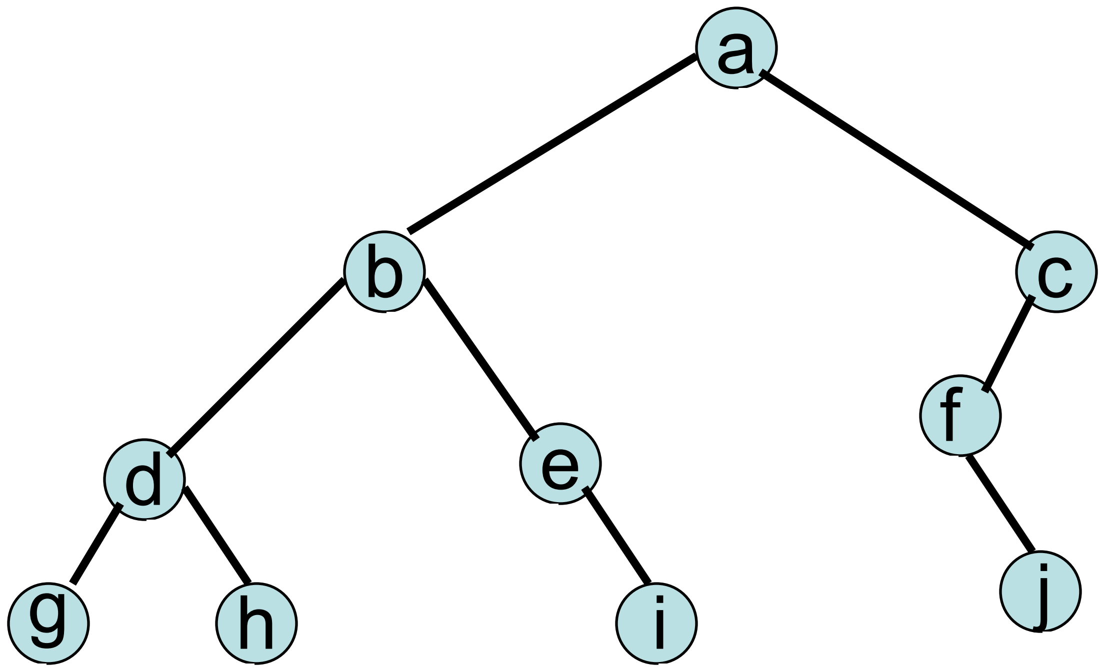
b a c

Inorder Example (visit = print)



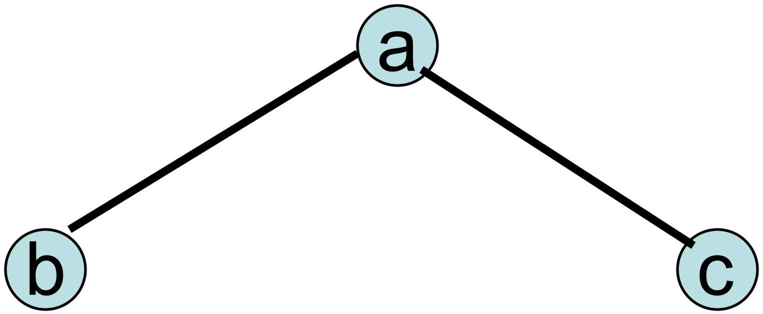
g d h b e i a f j c

Inorder By Projection (Squishing)



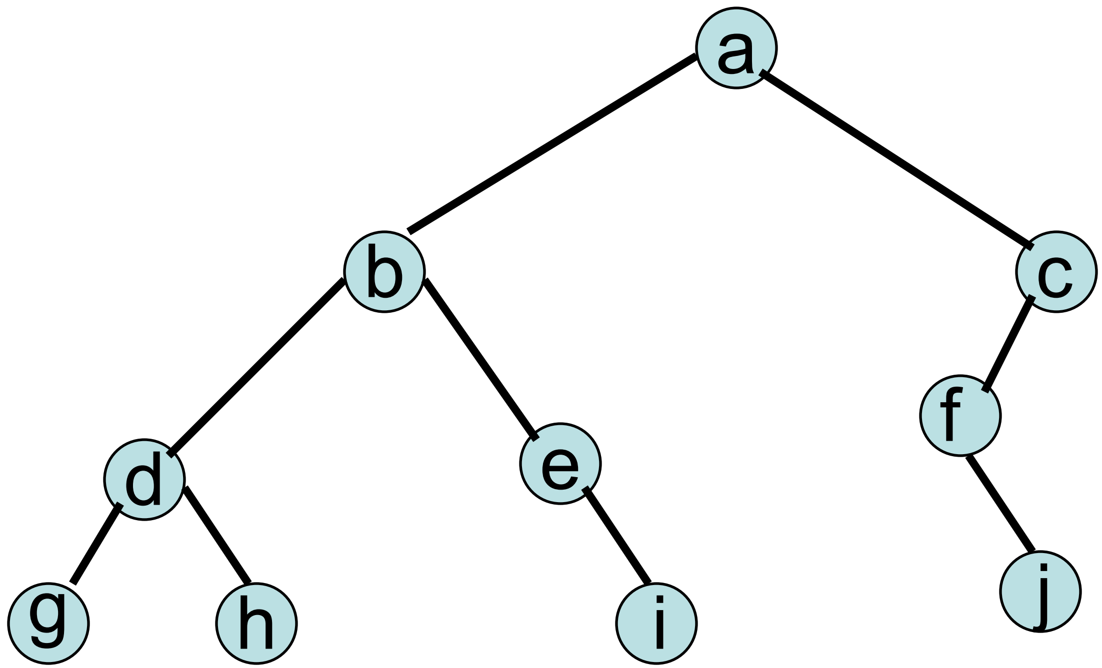
g d h b e i a f j c

Postorder Example (visit = print)



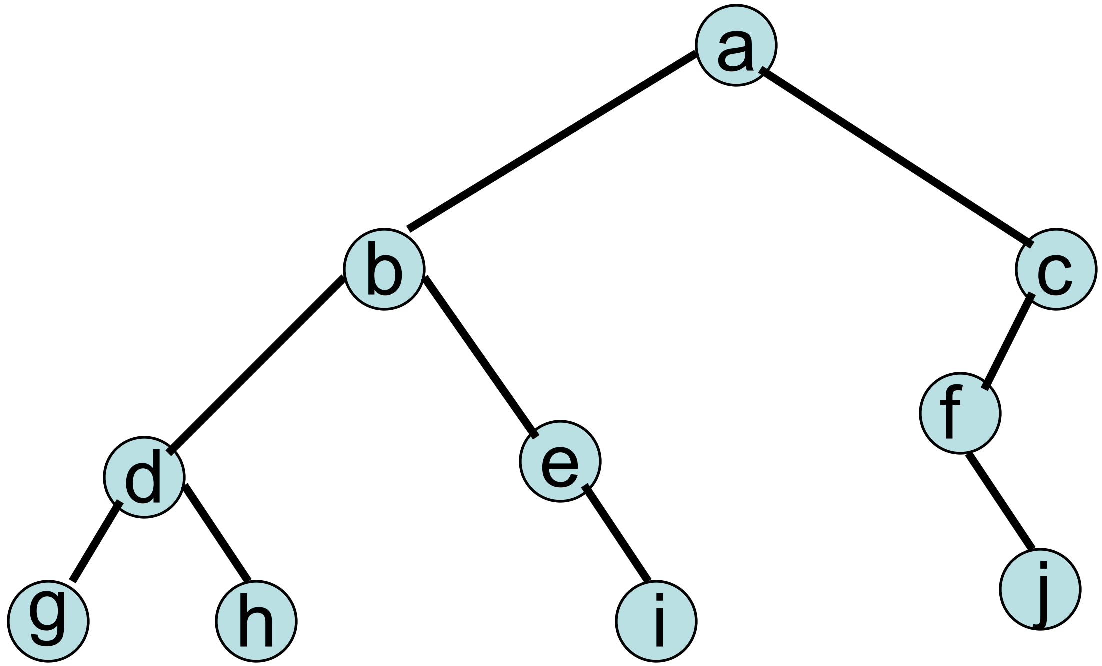
b c a

Postorder Example (visit = print)



g h d i e b j f c a

Traversal Applications



Pre
Construct
→ One order
→ 2 orders
In-order
left | Root | right
root | left | right

- Make a clone.

left right | root

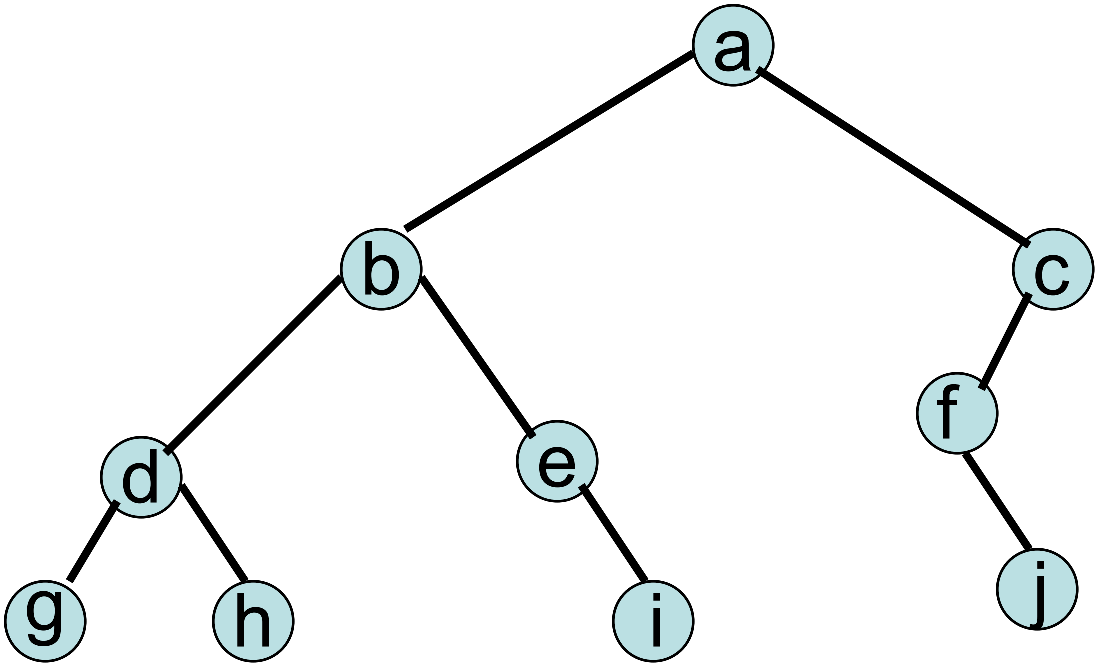
- Determine height.

left root right

- Determine number of nodes.

left | right | root
Binary Tree
Generalized

Level-Order Example



a b c d e f g h i j

Binary Tree Construction

- Suppose that the elements in a binary tree are distinct.
- Can you construct the binary tree from which a given traversal sequence came?
- When a traversal sequence has more than one element, the binary tree is not uniquely defined.
- Therefore, the tree from which the sequence was obtained cannot be reconstructed uniquely.

Binary Search Trees

- Dictionary Operations:
  - get(key)
  - put(key, value)
  - remove(key)

Complexity Of Dictionary Operations

Data Structure	Worst Case	Expected
Hash Table	O(n)	O(1)
Binary Search Tree	O(n)	O(log n)
Balanced Binary Search Tree	O(log n)	O(log n)

n is number of elements in dictionary

rarely

भारतीय प्रौद्योगिकी संस्थान जम्मू
INDIAN INSTITUTE OF TECHNOLOGY JAMMU

Thank You