CSI33 Data Structures

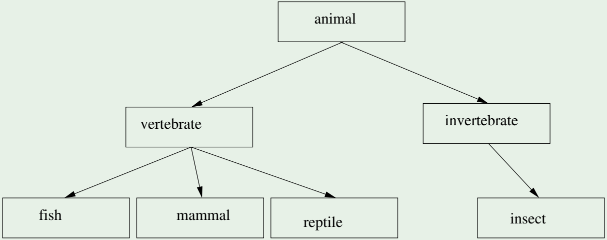
Department of Mathematics and Computer Science
Bronx Community College

Outline

1. Chapter 7: Trees
   - Tree Terminology
   - Example: Expression Trees
   - Binary Tree Representations
   - In-Class Work

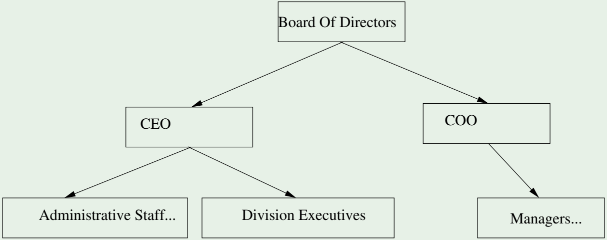
Uses Of Trees

Taxonomies



Uses Of Trees

Hierarchies



Uses Of Trees

Efficient Containers of Sequential Data

Each node represents a single data item of a collection organized, for efficient access, into a tree:

- Binary Search Trees
- Heaps
- Priority Queues
- B-Trees, Quad Trees, etc.

Definitions

Definitions I

- A tree consists of nodes connected by edges.
- A node can have zero or more child nodes. A child is connected to its parent node by a single edge.
- Children with the same parent are called siblings.
- A tree with no nodes or edges is an empty tree.
- A nonempty tree will have one special node with no parent called the root node.
- A node with no children is called a leaf.

Definitions

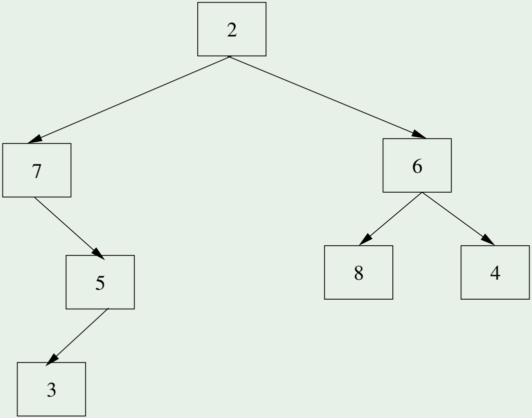
Definitions II

- All nodes are connected to the root by a path of edges.
- The depth of a node is the length of its path to the root. The root has depth zero.
- A level of a tree is a set of nodes which have the same depth.
- The descendants of a node are nodes whose paths include that node.
- The ancestors of a node are nodes on its path to the root.

Binary Trees

Binary Trees

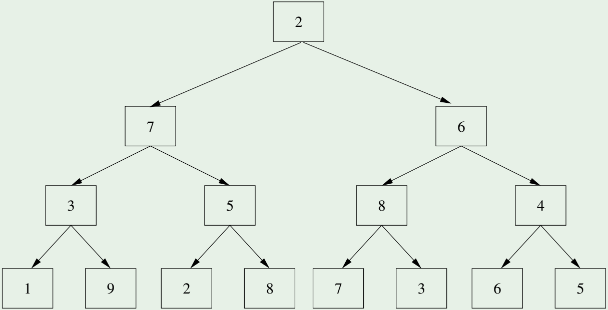
- A tree whose nodes have at most two children is a binary tree.



Binary Trees

Full Binary Trees

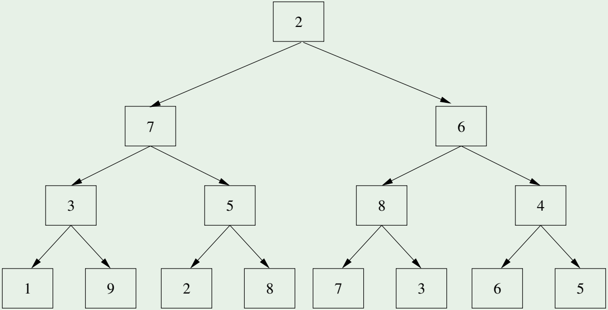
- A full binary tree is one whose levels have every position filled.



Binary Trees

Full Binary Trees

- A full binary tree is one whose levels have every position filled.

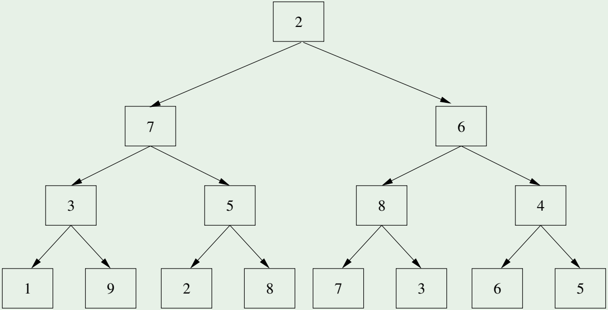


Question: how many nodes are in a full binary tree of height h?

Binary Trees

Full Binary Trees

- A full binary tree is one whose levels have every position filled.

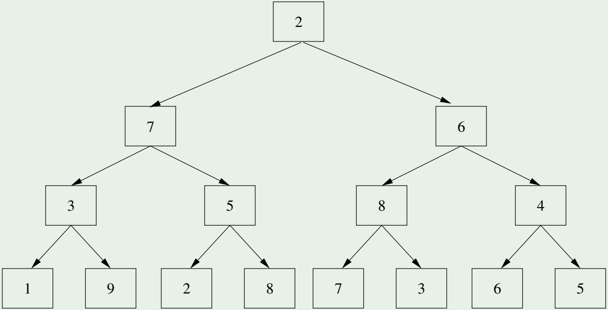


Question: how many nodes are in a full binary tree of height h? $2^h - 1$

Binary Trees

Full Binary Trees

- A full binary tree is one whose levels have every position filled.

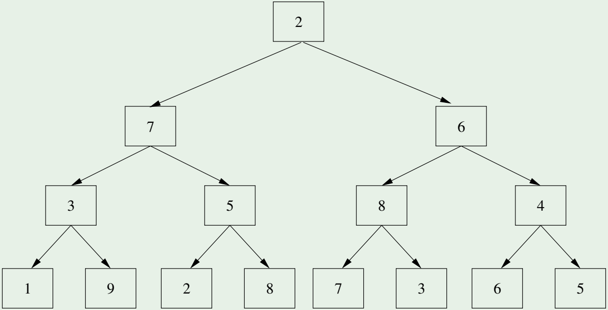


Question: how many leaves are in the full binary tree of height h?

Binary Trees

Full Binary Trees

- A full binary tree is one whose levels have every position filled.

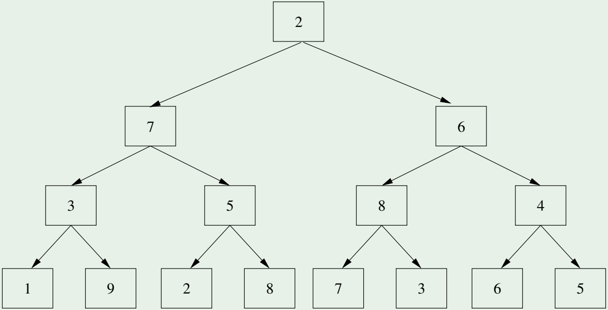


Question: how many leaves are in the full binary tree of height h? 2^{h-1}

Binary Trees

Full Binary Trees

- A full binary tree is one whose levels have every position filled.

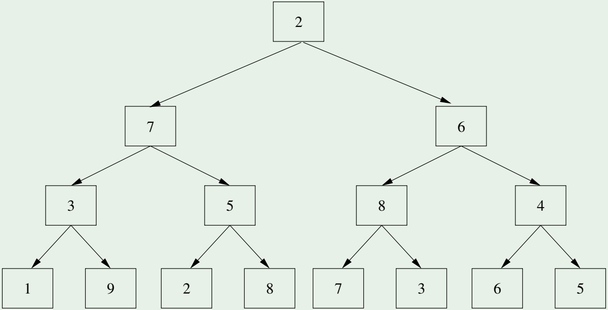


Question: If there are n nodes in a full binary tree, how many leaves are there?

Binary Trees

Full Binary Trees

- A full binary tree is one whose levels have every position filled.

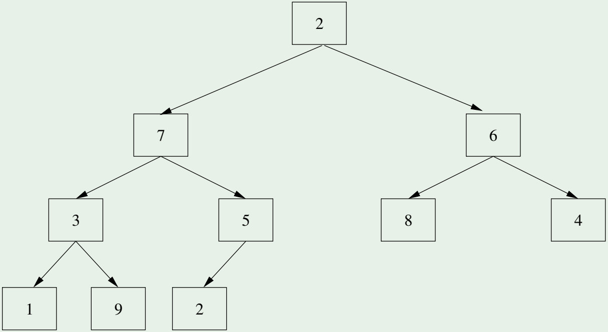


Question: If there are n nodes in a full binary tree, how many leaves are there? $\left\lceil \frac{n}{2} \right\rceil$

Binary Trees

Complete Binary Trees

- A complete binary tree has every level filled except the bottom, which is filled from left to right.



Binary Tree Traversal

Binary Trees are Recursive Data Structures

A binary tree can be defined as either

- An empty binary tree (base case), or
- a node having two binary trees as attributes, a left subtree and a right subtree (recursive case).

  Using this recursive definition, recursive algorithms can be used to process the nodes of a binary tree.

Binary Tree Traversal

Preorder

```
def traverse(tree):
   if tree is not empty:
      process data at tree's root
      traverse(tree's left subtree)
      traverse(tree's right subtree)
```

Binary Tree Traversal

Inorder

```
def traverse(tree):
   if tree is not empty:
      traverse(tree's left subtree)
      process data at tree's root
      traverse(tree's right subtree)
```

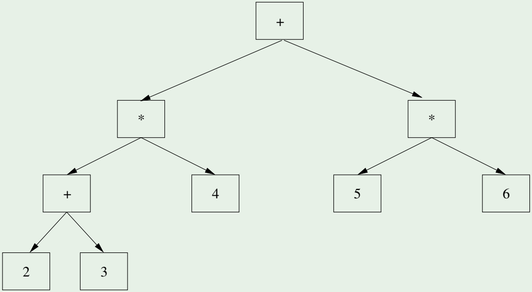
Binary Tree Traversal

Postorder

```
def traverse(tree):
  if tree is not empty:
    traverse(tree's left subtree)
    traverse(tree's right subtree)
    process data at tree's root
```

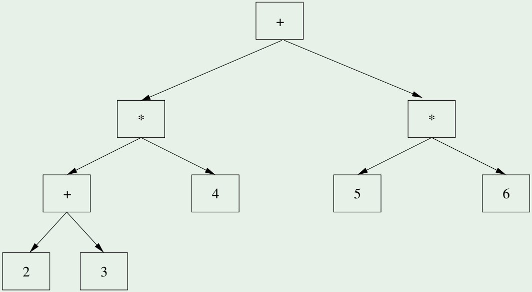
Binary Tree Representation Of An Expression

Abstract Syntax Tree



Binary Tree Representation Of An Expression

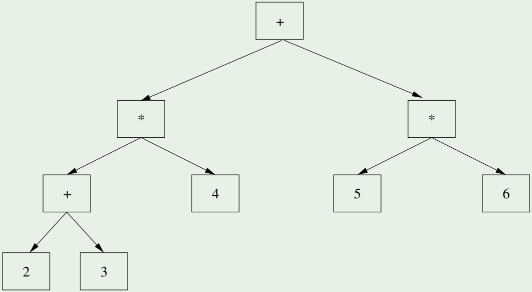
Different Traversals Give The Different Notations



- Printing in preorder gives Prefix Notation.
- Printing inorder gives Infix Notation.
- Printing in Postorder gives Postfix Notation.

Binary Tree Representation Of An Expression

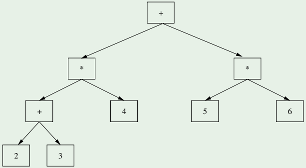
Different Traversals Give The Different Notations



- Printing in preorder gives Prefix Notation. +*+234*56
- Printing inorder gives Infix Notation. (2+3)*4+5*6
- Printing in Postorder gives Postfix Notation. 23+4*56*+

Binary Tree Representation Of An Expression

Evaluation




```
def evaluateTree(tree):
   if tree's root is an operand:
      return root data
   else:  # root contains an operator
      leftValue = evaluateTree(tree's left subtree)
      rightValue = evaluateTree(tree's right subtree)
      result = operator(leftValue, rightValue)
      return result
```

Linked Representation Of A Binary Tree

A TreeNode Class in Python

- An empty tree is represented by None.
- A nonempty tree is defined using a TreeNode as root.
- The TreeNode class is defined recursively.

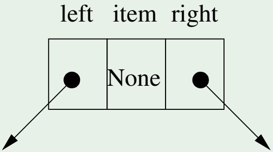
Linked Representation Of A Binary Tree

Recursive Definition Of A TreeNode Class

```
class TreeNode:
   def __init__(self, data = None, left=None, right=None):
      self.item = data
      self.left = left # TreeNode or None
      self.right = right # TreeNode or None
```

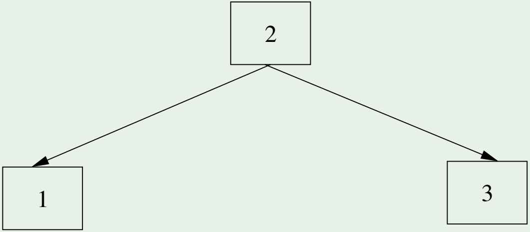
Python Examples Using The TreeNode Class

TreeNode



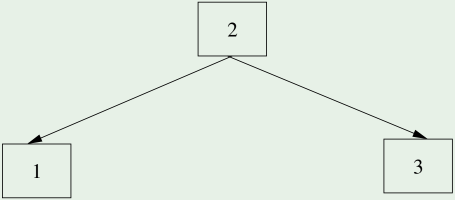
Linked Representation Of A Binary Tree

Example 1: Abstract View



Linked Representation Of A Binary Tree

Example 1: Abstract View

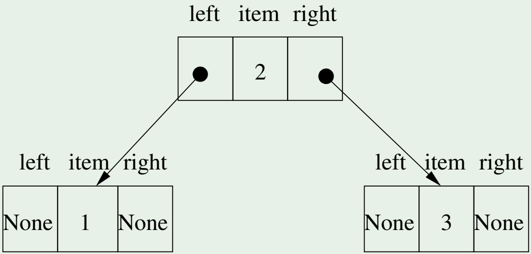


Example 1: Python Implementation

```
left = TreeNode(1)
right = TreeNode(3)
root = TreeNode(2, left, right)
```

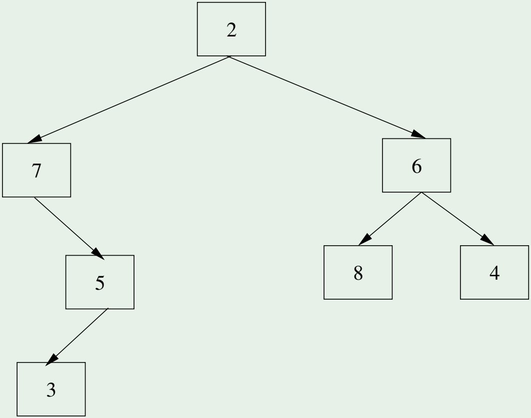
Linked Representation Of A Binary Tree

Example 1: Python Memory Model



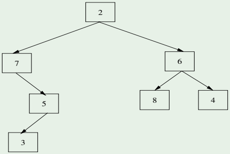
Linked Representation Of A Binary Tree

Example 2: Abstract View



Linked Representation Of A Binary Tree

Example 2: Abstract View

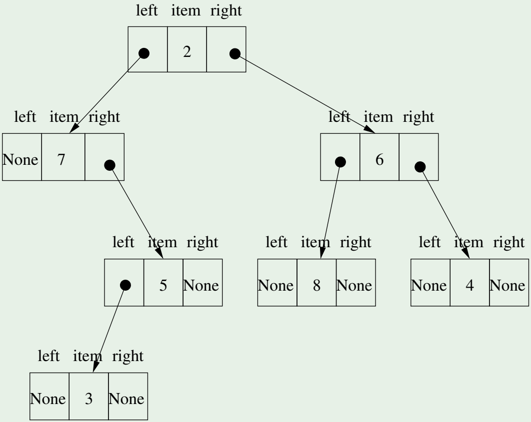


Example 2: Python Implementation

```
root = TreeNode(2,
              TreeNode(7,
                 None,
                 TreeNode(5, TreeNode(3))),
              TreeNode(6, TreeNode(8), TreeNode(4)))
```

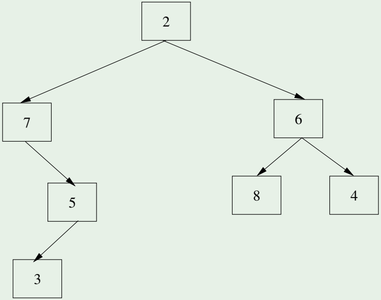
Linked Representation Of A Binary Tree

Example 2: Python Memory Model



Array Representation of Binary Trees

Abstract View - array representation



2	7	6	None	5	8	4	None	None	3
0	1	2	3	4	5	6	7	8	9

Array Representation of Binary Trees

Implementation

2	7	6	None	5	8	4	None	None	3
0	1	2	3	4	5	6	7	8	9

The node at position i has:

```
def left_child(i):
  return 2 * i + 1
def right_child(i):
  return 2 * i + 2
def parent(i):
  return (i - 1) // 2
```

In-Class Work

See the handout.