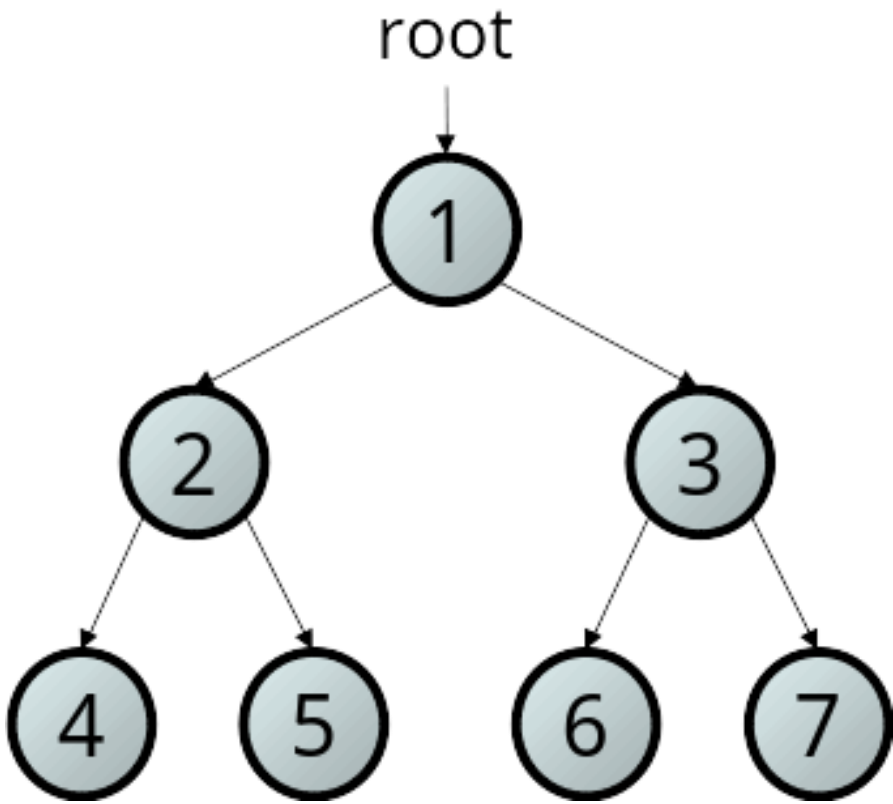
CSCI 241 Data Structures
Trees Intro

Western Red Cedar

WA DNR

Tree Form and Function

Vocabulary



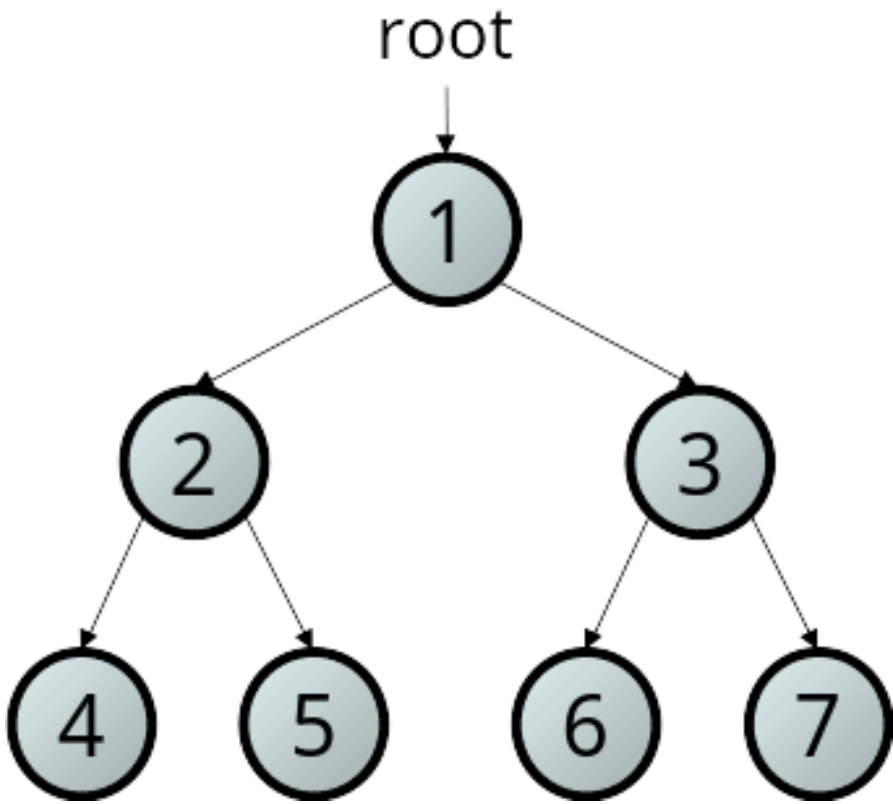
- **Node**: element of a tree that can hold data
- **Link**: a connection between two nodes

Note: so far, same idea as linked list!

- **Tree**: a directed, acyclic structure of linked nodes
  - **Directed**: one-way links between nodes
    - **Parent** and **child** nodes
  - **Acyclic**: no path wraps back around to the same node twice

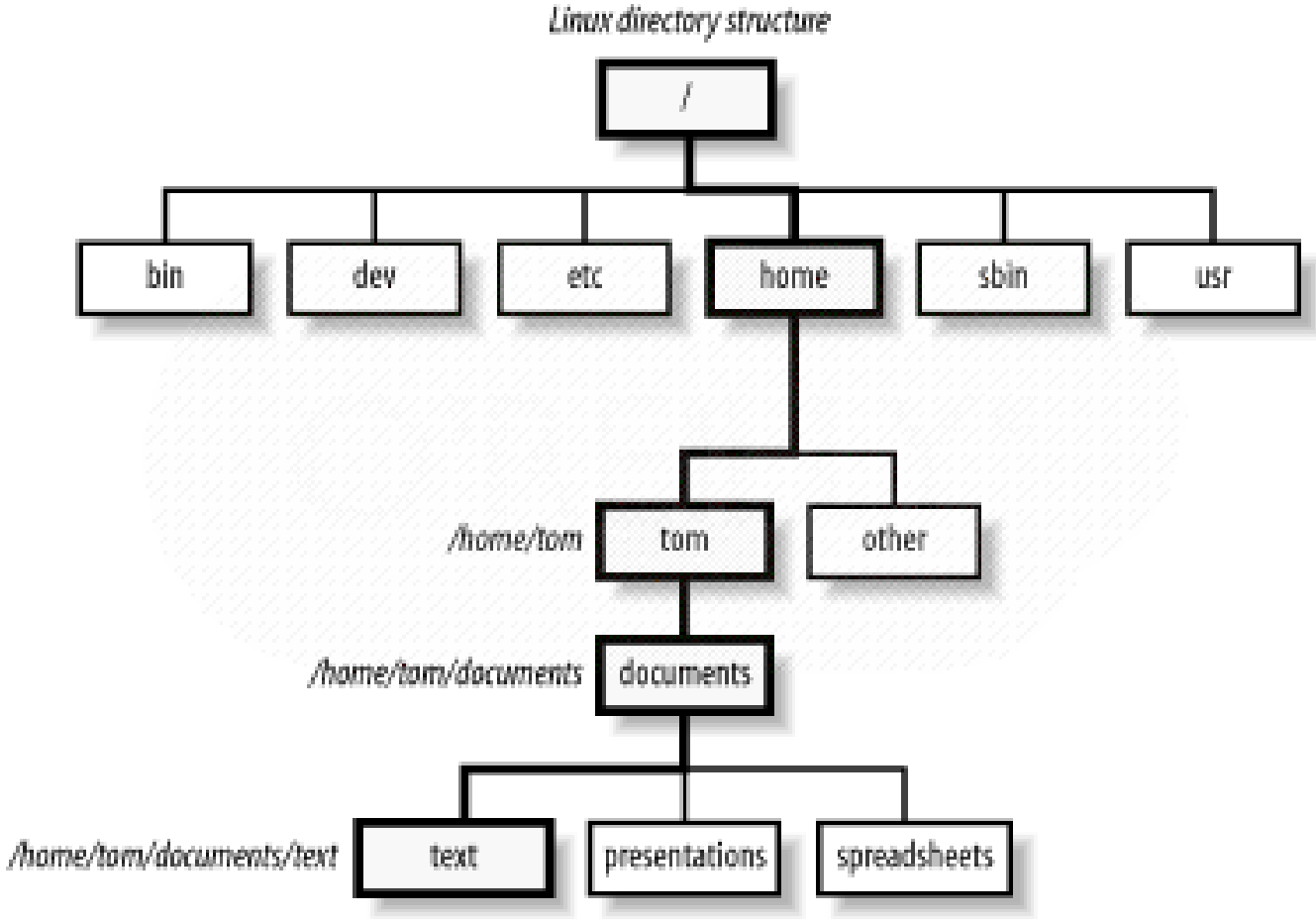
Vocabulary

- **Tree**: defined as either
  - Empty (null), or
  - A root node that contains:
    - Data
    - Possibly, links to child nodes
- Nodes with no children are **leaf nodes**
- **Root:** "top" node, no parents (usually only one)
- **Binary tree**: a tree where each node has at most two children.
  - **Left** and **right subtrees**



Examples

- Computer file structure
- Decision trees in AI
- Company org charts



Christophe Pallier

Binary Tree Properties

- **Height**: length of longest path from root to node
- **Level** or **depth**: length of path from root to a given node
- **Full tree**: one where every branch has two children

Example Implementation for Ints

```
// An IntTreeNode object is one node in a binary tree of ints.
public class IntTreeNode {
    public int data;            // data stored at this node
    public IntTreeNode left;    // reference to left subtree
    public IntTreeNode right;   // reference to right subtree









}
```

Example Implementation for Ints

```
// An IntTreeNode object is one node in a binary tree of ints.
public class IntTreeNode {
    public int data;            // data stored at this node
    public IntTreeNode left;    // reference to left subtree
    public IntTreeNode right;   // reference to right subtree

// Constructs a leaf node with the given data.
    public IntTreeNode(int data) {
        this(data, null, null);
    }









}
```

Example Implementation for Ints

```
// An IntTreeNode object is one node in a binary tree of ints.
public class IntTreeNode {
    public int data;            // data stored at this node
    public IntTreeNode left;    // reference to left subtree
    public IntTreeNode right;   // reference to right subtree

// Constructs a leaf node with the given data.
    public IntTreeNode(int data) {
        this(data, null, null);
    }

// Constructs a branch node with the given data and links.
    public IntTreeNode(int data, IntTreeNode left,
                                 IntTreeNode right) {
        this.data = data;
        this.left = left;
        this.right = right;
    }
}
```

Traversing Trees

NPR