Binary Trees

Read Sections 4.1 and 4.2
Binary Trees

B is the Left Child of A
B is the Parent of D
Right Child
Leaves
A **tree** is a collection of nodes. A tree is either empty or it contains a node called the **root** that is linked to zero or more subtrees. The subtrees of a node T are called the **children** of T and T is called the **parent** of the subtrees.

A **binary tree** is a tree where each node has at most 2 children.

A node with zero children is called a **leaf**.

A **path** in a tree is a sequence of nodes, $n_0, n_1, \ldots n_j$, where $n_i$ is the parent of $n_{i+1}$ for $0 \leq i < j$.

The **length** of a path is one less than the number of nodes in the path (this is equivalent to saying the length is the number of edges in the path).

The **height** of a tree is the length of the longest path from the root to a leaf. The height of an empty tree is -1. The height of a tree with only one node (the root) is 0.

The **depth** of a node, $n$, is the length of the path from the root to $n$. 
Binary Trees

B is the Left Child of A
B is the Parent of D

Height of the tree 3
Path from A to H: A, C, F, H
F is a depth 2
Binary Trees

Right subtree of A
Subtree rooted at C
Binary Trees

A **preorder** traversal of a binary tree rooted at node $n$, is a traversal where node $n$ is visited (the meaning of visited will vary based on the purpose of the traversal) followed by a preorder traversal of the left subtree of $n$ followed by a preorder traversal of the right subtree of $n$.

A **postorder** traversal of a binary tree rooted at node $n$, is a traversal where a postorder traversal of the left subtree of $n$ is completed followed by a postorder traversal of the right subtree of $n$ followed by a visit to $n$.

An **inorder** traversal of a binary tree rooted at node $n$, is a traversal where an inorder traversal of the left subtree of $n$ is completed followed by a visit to $n$ followed by an inorder traversal of the right subtree of $n$.

A **level** order traversal of a tree rooted at node $n$, is a traversal where the nodes in the tree are visited based on the depth of the node: $n$ is visited, followed by all nodes at depth 1, followed by all nodes at depth 2, ... until all the nodes at a depth equal to the height of the tree have been visited. A level order traversal is sometimes called a **breadth first** traversal.
## Binary Tree

### Preorder traversal
- If the tree is not empty
  - visit(root)
  - preOrder(left subtree)
  - preOrder(right subtree)

### Inorder traversal
- If the tree is not empty
  - inOrder(left subtree)
  - visit(root)
  - inOrder(right subtree)

### Postorder traversal
- If the tree is not empty
  - postOrder(left subtree)
  - postOrder(right subtree)
  - visit(Root)
Binary Trees

Preorder: A, B, D, E, C, F, H, G, I, J


Postorder: E, D, B, H, F, I, J, G, C, A
Binary Trees

Preorder: A, B, C

Inorder: B, A, C

Postorder: B, C, A
Full Binary Tree: A binary tree where every level has as many nodes as possible (i.e. every level is full).

Number of nodes in a full binary tree: $2^{h+1} - 1$
where $h$ is the height of the tree

Maximum number of nodes at depth $n$: $2^n$
Binary Trees

A complete binary tree is a binary tree in which every level but the deepest is full and the deepest level is either full or the nodes are as far left as possible.