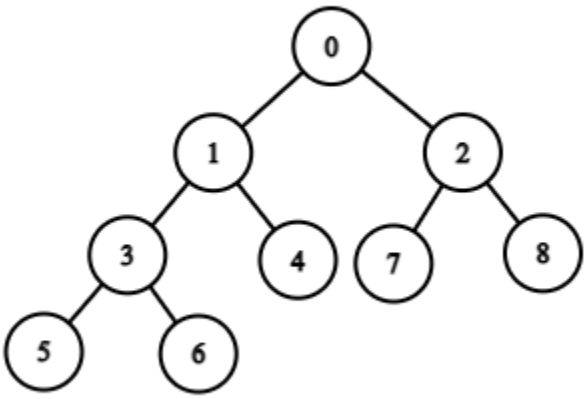
CSE 214 – Fall 2022

Recitation 06 – Binary Trees (Student Version)

1. **(5 minutes)** Consider the following tree:



1. What is the in-order traversal of this tree?

2. What is the pre-order traversal of this tree?

3. What is the post-order traversal of this tree?

2. (**5 minutes**) Given the following traversals, construct the general binary tree and list the post-order traversal.

In-order: 4 2 1 3 0 6 7 5

Pre-order : 0 1 2 4 3 5 6 7

3. **(5 minutes)** Can two or more different trees have the same pre-order and post-order? In other words, if you are given the pre-order and post-order of a binary tree, can you construct only one tree?

4. **(10 minutes)** Complete vs. Full Binary Trees
    1. A complete binary tree is a binary tree in which every level of the tree has the maximum number of nodes possible except possibly the deepest level. Also, at the deepest level, the nodes are as far left as possible. What are the formulas for the maximum and minimum number of nodes and internal nodes in a complete binary tree of height h.

    2. What is the minimum and maximum number of nodes in a complete binary tree with a height of 4? What is the number of internal nodes?

    2. A full binary tree is a binary tree in which every non-leaf node has 2 children, and all leaves have the same depth. What is the minimum and maximum number of nodes in a binary tree with a height of 4? What is the number of internal nodes? What is a full binary tree's relation to a complete binary tree?

5. **(15 minutes)** Assume we have the following classes declared:

```
public static class Node {
        int key;
        Node left;
        Node right;
        public Node(int key, Node left, Node right){
        this.key = key; this.left = left; this.right = right;
        }
}

public class Tree {
        Node root;
        public Tree(Node root){
        this.root = root;
        }
}
```

The Tree class is not a binary search tree because we will allow duplicates in this tree. Write a method count() for this class that counts the number of occurrences of a node with a given key in the tree. If the node is not in the tree, it returns 0.

6. **(15 minutes)** Create a mirror() method so that the left and right nodes are swapped at every level. Assume the node class has attributes left and right, and tree class has attribute root.

Example: