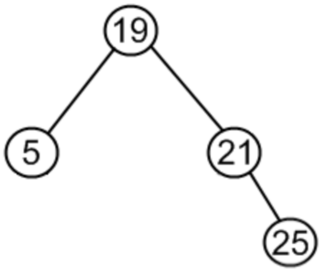
HW3 CSCD320

To Turn in: please put the questions and your answers below them in a pdf file for the question ONE. In addition, you have to zip up all your source code for the question TWO and your pdf file that contains your answer to question one, into a single zip file. Please turn in the single zip file on the **EWU Canvas** by going to CSCD320-01 course page on Canvas, then clicking Assignments→hw3->submit. Please name your zip file with your last name, followed by the initial of your first name, followed by hw3. For example, if you are John Smith, name your file as smithjhw3.zip

Background

Theory shows that the sequences of in-order traversal and post-order traversal determine a unique binary tree (this is a general binary tree and NOT limited to a BST). For example, given the sequence of the in-order traversal {5, 19, 21, 25} for a binary tree and given the sequence of post-order traversal {5, 25, 21, 19} for the same tree, you can reconstruct the binary tree with its graphical representation shown as below,



Question (1) to solve (70%)

Given the sequence of in-order traversal for a **general binary tree** {9, 5, 1, 7, 2, 12, 8, 4, 3, 11 }, and given the sequence of post-order traversal for the same tree { 9, 1, 2, 12, 7, 5, 3, 11, 4, 8}, please explain how to reconstruct the binary tree and draw its graphical representation in your answer. Please include step by step explanations to guarantee partial credits. Note that this tree is a general binary tree, but not a BST.

Hint: thinking about the definition of in-order and post-order traversal. For in-order traversal, after its left subtree is visited, then the root is visited. This is true for any subtrees of any node. For post-order traversal, the root is always outputted at last, after its left subtree and right subtree are both visited. This is true for any subtrees of any node in the tree.

Question (2) to solve (40%) (including a bonus of 10% on top of 100%)

Given the sequence of in-order traversal for a general binary tree, stored in an array **inOrderSequence[]=** {9, 5, 1, 7, 2, 12, 8, 4, 3, 11 }. And given the sequence of post-order traversal for the same tree, stored in an array **postOrderSequence[]=**{ 9, 1, 2, 12, 7, 5, 3, 11, 4, 8}. Please implement a **method** with its signature shown in the below,

BinaryTree buildTree(Object inOrderSequence[], Object postOrderSequence[])

-The method buildTree() returns a BinaryTree in memory that is constructed on the basis of the two input sequences — inOrderSequence and postOrderSequence.

-Please also implement the entire BinaryTree class and its nested Node class, in order to test your program. Please also implement the **in-orderTraversal()** and **post-orderTraversal()** method in your BinaryTree class to traverse the tree.

-Write a Tester class, in which you will call the method, like **BinaryTree mytree = buildTree(Object inOrderSequence[], Object postOrderSequence[]).**

-In the Tester class**,** then you call mytree. in-orderTraversal() and mytree. post-orderTraversal() methods to verify the tree returned by the buildTree() method is correct. If correct, you will see the output of mytree. in-orderTraversal() matches the input sequence **inOrderSequence[]** and the output of mytree. post-orderTraversal() matches the input sequence **postOrderSequence[].**