# CS221: Data Structures Binary Tree Homework

### John Ingram

## Problem 1:

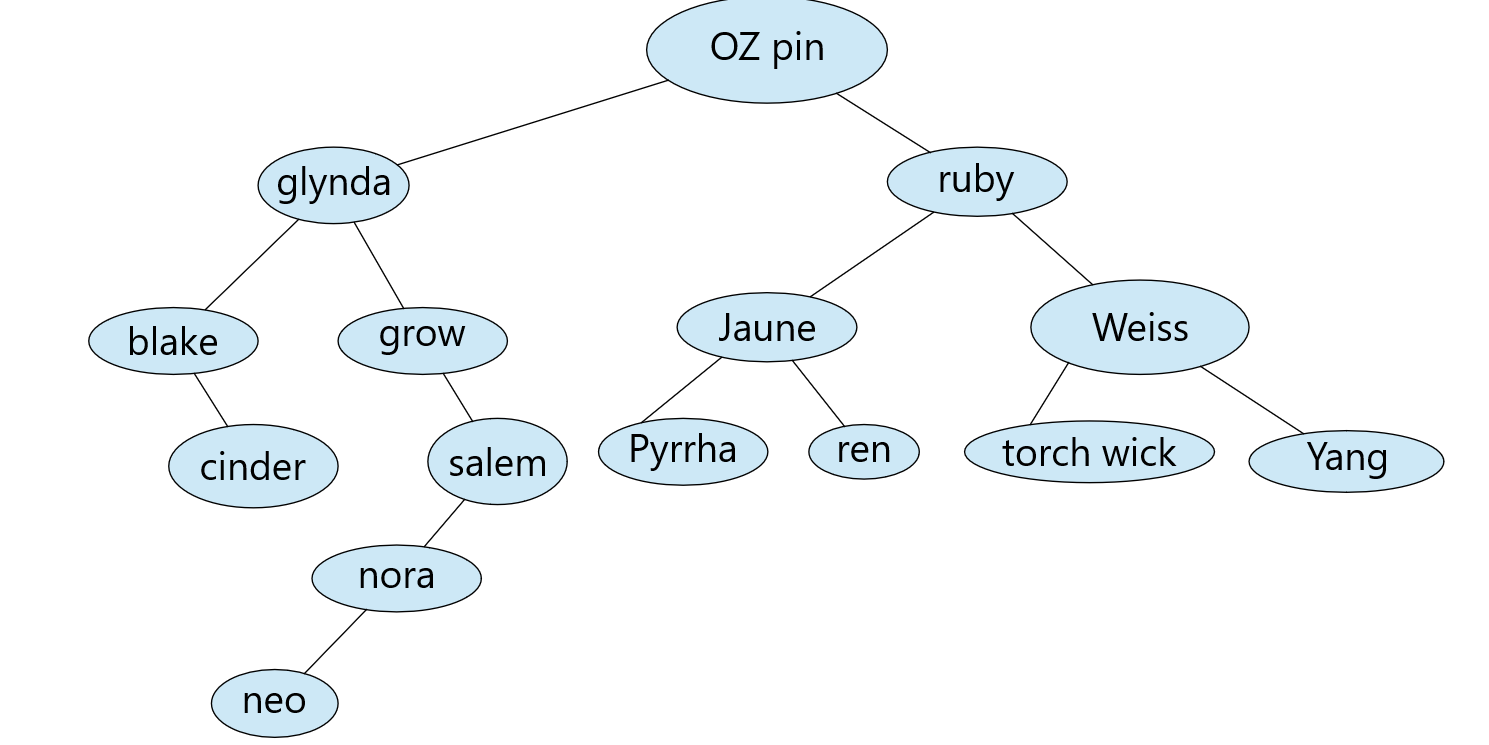
Consider the binary tree shown below. Show the order the nodes are visited in for each of the following traversal types:

* Preorder (52, 30, 11, 40, 32, 31, 38, 70, 61,68, 84, 81)
* Postorder (11, 31, 38, 32, 40, 68, 61, 81, 84, 70, 52)
* Inorder (11, 30, 31, 32, 38, 40, 52, 61, 68, 70, 81, 84)

## Problem 2:

Suppose we have a tree that can hold strings in the nodes. Draw the binary tree after the following sequence of string values (each separated by a comma) have been inserted into the tree. You may assume the sort order is a traditional alphabetical ordering.

ozpin, glynda, qrow, salem, ruby, weiss, blake, yang, jaune, nora, pyrrha, ren, torchwick, neo, cinder



## Problem 3:

Write a solution to return the count of the number of nodes in a binary tree. Your method will be passed one parameter, a copy of a pointer to the root node of the tree (Node \*) and will return an int that is the count of nodes. If the tree is empty, return 0 (this is the recursive base case).

int nodeCount(Node \*root)

{

    if (root == NULL) return 0;

    return 1 + nodeCount(root->left) + nodeCount(root->right);

}

## Problem 4: Challenge problem, not graded.

Write code to display the nodes of a tree starting at the root, and, use indentation and labeling to show the structure of the tree. For example, if our tree is:

The display should look something like:

5

L: 3

L: 1

L: null

R: null

R: 4

R: 8

L: null

R: null