Homework 2

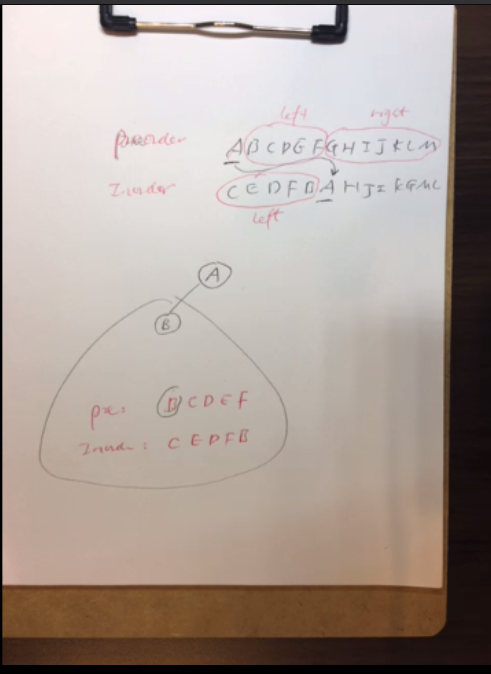
1: Attached code. Used code cited from <https://www.geeksforgeeks.org/iterative-program-count-leaf-nodes-binary-tree/>

2:

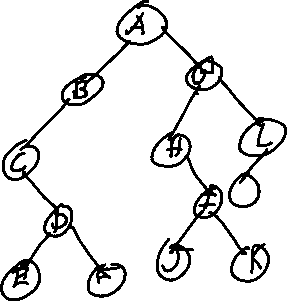
Let is any integer element of both set and such that represents the values of nodes on a binary tree in Inorder sequence and represents the value of nodes on a binary treen in Preorder sequence.   
To find ’s location on a binary tree, , we must find its height on said tree, , its parent, , and if it is the left or right subtree of that parent.

We may find the height of by using both our sequences. By definition of preorder, the leading value of the preorder set, , will always be the root of the tree. We also know that exists in the postorder sets, since by definition all elements of the postorder set are in the preorder set, and visa versa. Knowing this, we can find that our left and right subtrees of this root are the immediate successor and predecessor of in our postorder set, by definition of postorder. Knowing these values, which we will name , we may find our subtrees of said values using our preorder set. By definition of preorder, after our value , a preorder set will contain that node’s left subtree followed that node’s right subtree. We know the child node of

Using preorder you can find the root of the tree, then using inorder you can find the immediate left and right children. Using how preorder is structured, you can then find the left and right subtrees of that parent node. You can then recurse though the list to find the tree’s exact shape.

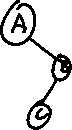
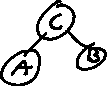
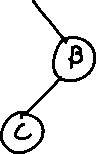
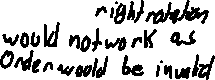
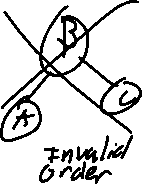


3:



4:

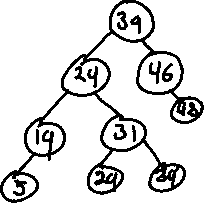
5:



6:

The balance factor of the root node is 0.

7:



8:

The complexity of constructing a binary tree will always be . This can be determined entirely based on the context that sorting takes as if constructing was faster, we could simply construct a new, sorted list in less than , and since means the best case complexity, this is impossible.

. 9:

10:

11:

12:

13:

14:

1 – 1

2 – 0

3 – 0

4 – 2

5 – 10

6 – 4

7 – 40

8 – 92

9 – 352

10 – 724