Jacob Sriraman

Problem #3 Binary Search Trees

A way to make a more balanced binary tree would be to implement something similar to an AVL tree, that can “balance itself”, by ensuring that the tree remains balanced after each insertion or deletion, by rotating when balance needs to be restored. Something like that could be done with the pseudocode below:

Def insert(root, key)

If not root:

Return Node(key)

If key < root.key:

Root.left = insert(root.left, key)

Else:

Root.right = insert(root.right, key)

// Update height and balance factor

Height = 1 + (right.height + left.height)

Def balance(node):

Def getheight(node):

Def getbalancefactor(node)

And lastly some function for rotatations.

This will be demonstrated in my AVL tree code.