1. private void updateHeight(TreeNode< T > root);

   - Parameters:
     root: Represents the number which is the root of the tree.
   - Return value:
     Does not have a return value.

2. private int getBalance(TreeNode< T > root);

   - Parameters:
     root: takes in the number which is the root of the tree and sees if it is balanced.
   - Return value:
     Returns and integer, probably between -2 and 2 to see if it is balanced. If greater or less than 1/ -1 respectively then not balanced.

3. private TreeNode< T > rebalance(TreeNode< T > root);

   - Parameters:
     root: Represents the number which is the root of the tree.
   - Return value:
     TreeNode¡T¿: returns the tree node that would balance the tree.

4. private TreeNode< T > rightRotate(TreeNode< T > root);

   - Parameters:
     root: Represents the number which is the root of the tree.
   - Return value:
     Will return the tree rotated to the right, creating a new root for the tree.

5. What is the formula for computing the height of a tree's root node, assuming the heights for its left and right subtrees are known?

   The formula for computing the height of a tree's node is: $height = Max(leftchild, rightchild) + 1$

6. What should the height be for a newly allocated leaf node? For an empty subtree?

   The height for a newly allocated leaf node is 0 because their is nothing below it. An empty subtree counts as -1 for balancing purposes.

7. Why do the insert() and remove() methods declare recursive helper functions, rather than themselves being recursive?

   They do that so it can return the new root node of the tree for the actual insert and remove methods.

8. Here is one line from the insertHelper() method:
   root.setLeft(insertHelper(value, root.left));

Explain why the root's child pointer must be updated with the result of the call to insertHelper().

The root's child pointer must be updated because when inserting a new node it may change the structure of the tree causing it to rotate left or right. So certain nodes children may need to be updated.