Example 2: A Complex Rotation

BST Rotation Summary:
1. Four kinds of rotations (L, R, LR, and RL)
2. All rotations are local
3. All rotations run in constant time, O(1)
4. BST property is maintained!

Overall Goal:
...and we call these trees:

...additional property:

AVL Theorem #1: If an insertion occurred in subtrees \( t_3 \) or \( t_4 \) and a subtree was detected at \( t \), then a _________ rotation about \( t \) restores the balance of the tree.

AVL Theorem #2: If an insertion occurred in subtrees \( t_2 \) or \( t_3 \) and a subtree was detected at \( t \), then a _________ rotation about \( t \) restores the balance of the tree.
AVL Insertion

Pseudocode:

```
struct TreeNode {
    T key;
    unsigned height;
    TreeNode *left;
    TreeNode *right;
}
```

AVL Insertion

```
template <class T>
void AVLTree<T>::_insert(const T & x, TreeNode<T> * & t) {
    if( t == NULL ) {
        t = new TreeNode<T>(x, 0);
    }

    else if( x < t->key ) {
        _insert( x, t->left );
        int balance = height(t->right) - height(t->left);
        int leftBalance = height(t->left->right) - height(t->left->left);
        if ( balance == -2 ) {
            if ( leftBalance == -1 ) { rotate________( t ); }
            else { rotate________( t ); }
        }
    }

    else if( x > t->key ) {
        _insert( x, t->right );
        int balance = height(t->right) - height(t->left);
        int rightBalance = height(t->right->right) - height(t->right->left);
        if ( balance == 2 ) {
            if ( rightBalance == 1 ) { rotate________( t ); }
            else { rotate________( t ); }
        }

        t->height = 1 + max(height(t->left), height(t->right));
    }
}
```

AVL Removal

CS 225 – Things To Be Doing:

1. Theory Exam 2 is ongoing
2. MP4 released today; due Monday, March 12
3. lab_huffman is due Sunday, March 4
4. Daily POTDs