Four AVL Rotation Templates:

Detecting Imbalance:

b =

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.

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:

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

AVL.cpp (snippet)

```cpp
template <class T>
void AVLTree<T>::_insert(const T & x, treeNode<T> * & t) {
  if ( t == NULL ) {
    t = new TreeNode<T>(x, 0, NULL, NULL);
  } 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 ); }
    }
  } else if ( x == t->key ) return;
  t->height = 1 + max(height(t->left), height(t->right));
}
```

AVL Removal

AVL Analysis

We know: ___________________.

We will argue: h = ________________.

Big-O is defined as:

Visually:

CS 225 – Things To Be Doing:

1. Exam #6 live now! (Programming exam: lists, trees)
2. MP4 extra credit submission starts tonight!
3. New lab on Wednesday
4. Daily POTDs