Insertion into an AVL Tree

Insert (pseudo code):
1: Insert at proper place
2: Check for imbalance
3: Rotate, if necessary
4: Update height

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

_\text{insert}(6.5)_
template <typename K, typename V>
void AVL<K, D>::_insert(const K & key, const V & data, TreeNode * & cur) {
    if (cur == NULL) {
        cur = new TreeNode(key, data);
    } else if (key < cur->key) { _insert(key, data, cur->left); }
    else if (key > cur->key) { _insert(key, data, cur->right); }
    _ensureBalance(cur);
}
template <typename K, typename V>
void AVL<K, D>::_ensureBalance(TreeNode *& cur) {
    // Calculate the balance factor:
    int balance = height(cur->right) - height(cur->left);

    // Check if the node is currently not in balance:
    if (balance == -2) {
        int l_balance =
            height(cur->left->right) - height(cur->left->left);
        if (l_balance == -1) { ________________; }
        else { ________________; }
    } else if (balance == 2) {
        int r_balance =
            height(cur->right->right) - height(cur->right->left);
        if (r_balance == 1) { ________________; }
        else { ________________; }
    }

    _updateHeight(cur);
};
AVL Tree Analysis

We know: insert, remove and find runs in: ___________.

We will argue that: h is ___________.

AVL Tree Analysis

Definition of big-O:

...or, with pictures:

\[ n, \text{ number of nodes} \]
AVL Tree Analysis
AVL Tree Analysis

- The number of nodes in the tree, $f^{-1}(h)$, will always be greater than $c \times g^{-1}(h)$ for all values where $n > k$. 
Plan of Action

Since our goal is to find the lower bound on $n$ given $h$, we can begin by defining a function given $h$ which describes the smallest number of nodes in an AVL tree of height $h$: 
Simplify the Recurrence

\[ N(h) = 1 + N(h - 1) + N(h - 2) \]
**State a Theorem**

**Theorem:** An AVL tree of height \( h \) has at least ________.

**Proof:**

I. Consider an AVL tree and let \( h \) denote its height.

II. Case: ______________

An AVL tree of height ____ has at least ____ nodes.
Prove a Theorem

III. Case: ______________

An AVL tree of height _____ has at least _____ nodes.
Prove a Theorem

By an Inductive Hypothesis (IH):

We will show that:

An AVL tree of height ____ has at least ____ nodes.
Prove a Theorem

V. Using a proof by induction, we have shown that:

...and inverting: