#19: AVL Analysis

March 2, 2022 · *G Carl Evans*

AVL Insertion

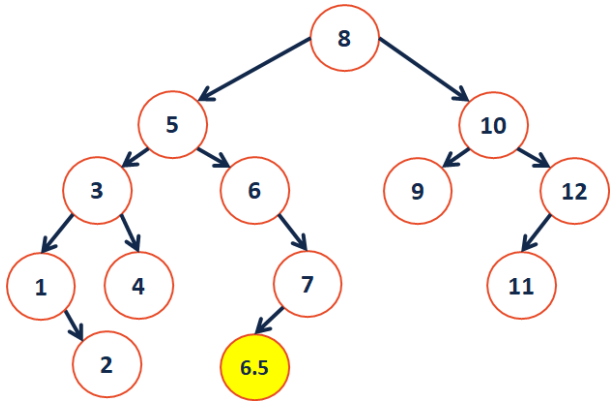
AVL.h (snippet)

```
23  class TreeNode {
24    public:
25      T key;
26      unsigned height;
27      TreeNode *left;
28      TreeNode *right;
 …
```

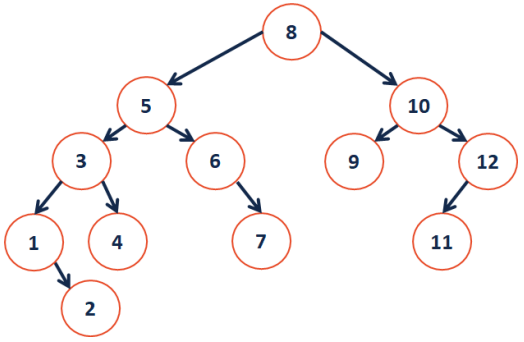
AVL.hpp

```
151 template <typename K, typename V>
152 void AVL<K, D>::_insert(const K & key, const V & data, TreeNode
    *& cur) {
153   if (cur == NULL)          { cur = new TreeNode(key, data);   }
157   else if (key < cur->key) { _insert( key, data, cur->left ); }
160   else if (key > cur->key) { _insert( key, data, cur->right );}
166   _ensureBalance(cur);
167 }
---
119 template <typename K, typename V>
120 void AVL<K, D>::_ensureBalance(TreeNode *& cur) {
121   // Calculate the balance factor:
122   int balance = height(cur->right) - height(cur->left);
123
124   // Check if the node is current not in balance:
125   if ( balance == -2 ) {
126     int l_balance =
            height(cur->left->right) - height(cur->left->left);
127     if ( l_balance == -1 ) { _____________________________; }
128     else                   { _____________________________; }
129   } else if ( balance == 2 ) {
130     int r_balance =
            height(cur->right->right) - height(cur->right->left);
131     if( r_balance == 1 ) { _____________________________; }
132     else                 { _____________________________; }
133   }
134
135   _updateHeight(cur);
136 };
```

AVL Insertion

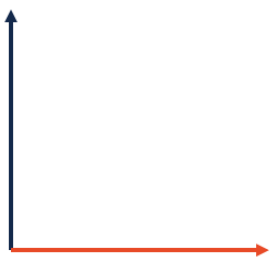


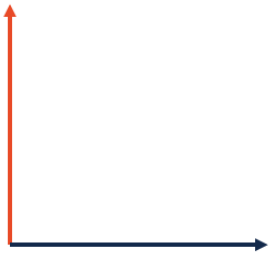
AVL Removal



Motivation:
Big-O is defined as:

Let **f(n)** describe the height of an AVL tree in terms of the number of nodes in the tree (**n**). Visually, we can represent the big-O relation:





$f(n) \leq c \times g(n)$: Provides an <u>upper bound</u>:

The height of the tree, **f(n)**, will always be <u>less than</u> **$c \times g(n)$** for all values where **$n > k$**.

$f^{-1}(h) \geq c \times g^{-1}(h)$: Provides a <u>lower bound</u>:

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

Plan of Action:
Goal: Find a function that defines the lower bound on **n** given **h**.

Given the goal, we begin by defining a function that describes the smallest number of nodes in an AVL of height **h**:

Theorem:
An AVL tree of height **h** has at least _____.

I. Consider an AVL tree and let **h** denote its height.

II. Case: _____

III. Case: _____

Inductive hypothesis (IH):

Proving our IH:

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

...and by inverting our finding:

CS 225 – Things To Be Doing:
1. Decide on Final Project or Final Exam and form groups **2.** mp_mosaics ongoing! **3.** Daily POTDs are ongoing!