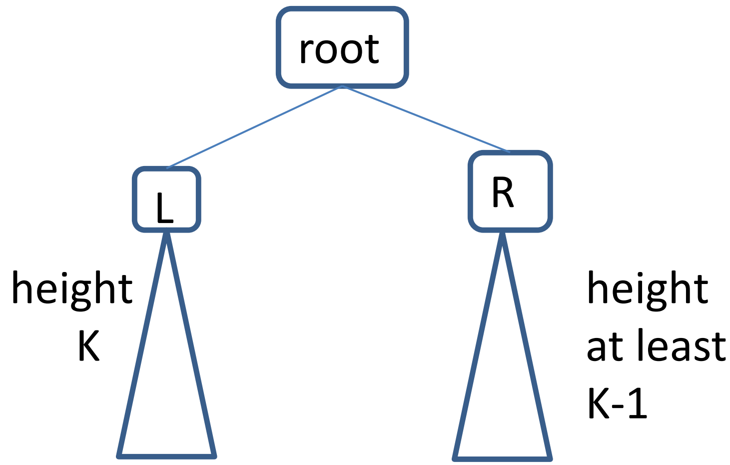
Heights of AVL Trees

How do we know AVL trees have O(log(n)) operations?

Theorem: An AVL tree with height H has at least Fib(H+3)-1 nodes, where Fib( ) is the Fibonacci function -- Fib(0)=0, Fib(1)=1, Fib(2)=1, Fib(3) = 2, Fib(4) = 3, Fib(5) = 5, Fib(6)=8, etc.
Proof: This is true for small values of H. If H=0 the tree has one node, and H(3)-1 = 1. If H=1 the tree must have at least 2 nodes and Fib(4)-1 = 2.

This is the base case. We'll do the inductive case on the next slide.

Suppose the theorem is true for H = 0, 1, 2, ... K and we have an AVL tree with height K+1. It must look like this:



One of the subtrees of the root must have height K; the other must have height at least K-1 or the root won't satisfy the AVL property. By the inductive hypothesis the taller subtree has at least Fib(K+3)-1 nodes and the smaller at least Fib(K+2)-1 nodes.

Altogether we have 1 (for the root) + Fib(K+3)-1 + Fib(K+2)-1 nodes. This simplifies to Fib(K+3)+Fib(K+2)-1, which is Fib(K+4)-1. This is what the theorem says for H = K+1. So if the theorem is true for all of the numbers up to K, it must be true for K+1 as well. This means it is true for all numbers.

Now, how big if Fib(H+3)-1?? One can show that

Fib(n) = $\frac{A^n - B^n}{\sqrt{5}}$ where A= $\frac{1+\sqrt{5}}{2} \approx 1.62$ and B= $\frac{1-\sqrt{5}}{2} \approx -0.62$

For large values of n B^n will be near 0, so Fib(n) is very close to (A^n)

Our theorem says that the number of nodes is exponential in the height, so the height is logarithmic in the number of nodes, i.e.

$$\text{Height} = O(\log(n))$$

Since all of our operations are implemented by walking from the root to a leaf and back, this means that find(), insert() and remove() are all O(log(n)) operations in an AVL tree.