Alin Tomescu | Week 1, Wednesday, March 5th, 2014 | Recitation 8
6.006 Intro to Algorithms | Prof. Srinivas Devadas | Prof. Nancy Lynch | Prof. Vinod Vaikuntanathan

AVL trees height proof

Let $N(h)$ denote the **minimum** number of nodes in an AVL tree of height h. Let r denote the root node of this tree.

Remember: A single-node tree has height 0, and a complete binary tree on $n + 1$ levels has height n. See figure below:

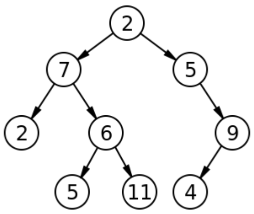
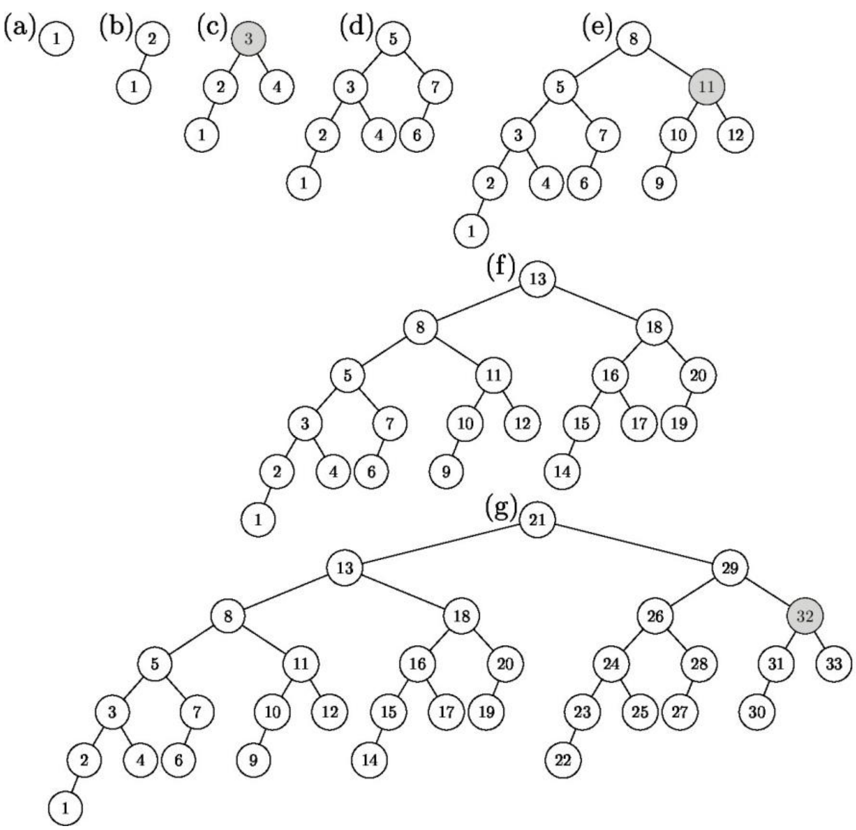


Figure 1: A simple binary tree of size 9 and height 3, with a root node whose value is 2. The above tree is unbalanced and not sorted.

Note that AVL trees with a *minimum* number of nodes are *the worst case examples* of AVL tree: every node's subtrees differ in height by one. You can see examples of such trees below:



If we can bound the height of these worst-case examples of AVL trees, then we've pretty much bounded the height of all AVL trees.

Note that we cannot make these trees any worse / any more unbalanced. If we add a leaf node, we either get a non-AVL tree or we balance one of the subtrees, which we don't want. If we remove a leaf node, we either get a non-AVL tree or we balance one of the subtrees.

Observation 1: If the AVL tree rooted at r has a minimum number of nodes, then one of its subtrees is higher by 1 than the other subtree. Otherwise, if the two subtrees were equal, then the AVL tree rooted at r is not minimal: we can always make it smaller by removing a few nodes from one of the subtrees and making the height difference ± 1.

Assume, without loss of generality, that the left subtree is bigger than the right subtree. We can express $N(h)$ in terms of:

- $N(h-1)$, the minimum number of nodes in the left subtree of r
- $N(h-2)$, the minimum number of nodes in the right subtree of r.

$$N(h) = 1 + N(h-1) + N(h-2)$$

We assumed that $N(h-1) > N(h-2)$, so we can say that

$$N(h) > 1 + N(h-2) + N(h-2) = 1 + 2 \cdot N(h-2) > 2 \cdot N(h-2)$$

So we have:

$$N(h) > 2 \cdot N(h-2)$$

We can try to solve this as a recurrence (note that $N(0) = 1$):

$$N(h) > 2 \cdot N(h-2) > 2 \cdot 2 \cdot N(h-4) > 2 \cdot 2 \cdot 2 \cdot N(h-6) > \cdots > 2^{h/2}$$

You can see it's $2^{h/2}$ by checking for a particular $h = 6$:

$$N(6) > 2 \cdot N(6-2) > 2 \cdot 2 \cdot N(4-2) > 2 \cdot 2 \cdot 2 \cdot N(2-2) > 2^3$$

Now, we can try and bound h:

$$N(h) > 2^{h/2} \overset{Take \log}{\Longleftrightarrow} \log N(h) > \log 2^{h/2} \Leftrightarrow h < 2 \log N_h$$

Thus, these worst-case AVL trees have height $h = \mathrm{O}(\log n)$.

This means that nicer / more balanced AVL trees will have the same bound on their height. You can think of such trees as worst-case trees with some of the missing nodes "filled in."