### Today
- AVL Trees

### Assignments
- HW9 due
- Exam 2 out (due next Tues before class)

Recall the definitions of “full”, “complete”, and “balanced”

- **Full**: (i) internal nodes have 2 children, (ii) ea. node’s subtrees same height
- **Complete**: full at level $n - 1$, level $n$ filled from left to right
- **Balanced**: ea. node’s left and right subtree heights differ by at most 1
What about the height for balanced binary trees?

They might not have minimum height (like in complete trees) ...

However, the height of a balanced binary tree is still $O(\log n)$ ...

- To show this, given a tree with $n$ nodes, determine the max tree height $h$
- Which is the same as finding the min number of nodes $n$ in a tree of height $h$

Basic idea:

- Let $T_h$ be a balanced binary tree of height $h$ with the minimum possible nodes
  - with $T_l$ and $T_r$ the left and right subtrees of $T_h$
- It must be that:
  - $T_l$ or $T_r$ has height $h - 2$, and the other height $h - 1$
  - we'll assume $T_l$ has height $h - 1$ and $T_r$ has height $h - 2$
- For $|T|$ the number of nodes in a tree $T$, this means that:
  $$|T_h| = |T_{h-1}| + |T_{h-2}| + 1$$
- These are “fibonacci trees” since they follow the fibonacci sequence
Some examples:

- \( T_0 = 0 \) nodes
- \( T_1 = 1 \) node
- \( T_2 = 2 \) nodes
- \( T_3 = 4 \) nodes
- \( T_4 = 7 \) nodes
- \( T_5 = 12 \) nodes

Assuming \( F_i \) is the \( i \)-th fibonacci number ...

\[
F_0 = 0, \ F_1 = 1, \ F_2 = 1, \ F_3 = 2, \ F_4 = 3, \ F_5 = 5, \ F_6 = 8, \ F_7 = 13, \text{ etc}
\]

Where \( |T_h| = F_{h+2} - 1 = n \), s.t. \( n \) is the min number of nodes at height \( h \)

- The \( i \)-th fibonacci number is approximated\(^1\) to be \( F_i \approx 2^{0.694i} \)
- Thus, \( n = F_{h+2} - 1 \approx 2^{0.694(h+2)} - 1 \)

After solving for \( h \), we have that the (worst case) height is still \( O(\log n) \)

\(^1\)See *Algorithms*, Dasgupta et al., pg 3, 2008
AVL Trees

Basic Idea: (animated: \url{www.cs.usfca.edu/~galles/visualization/AVLtree.html})

- Calculate a “balance factor” for each node (left subtree – right subtree height)
  - Note: just store the height of ea. subtree root and calculate balance factor
- After every insert and delete, do “tree rotations” if needed
  - Either a single or double rotation may be needed
- Apply rotations (deletion) and update balance factors “up the tree”

Note: parent becomes 4’s right-child, and 4’s right-child becomes parent’s left-child