CSCI 241
Lecture 14c
AVL removal, runtime
Goals

• Know how to remove nodes from AVL trees.

• Know the runtime of searching, inserting, and removing in AVL trees.
Removing from AVL Tree

• Much like insertion: remove as usual, rebalance as necessary at each level up to the root.

• Whereas insertion only ever requires only one rebalance, deletion can require many
  • but still no more than the tree’s height.
Runtime in AVL Trees

- As usual, runtime of search, insert, and remove are all $O(\text{height})$.

- A rotation is $O(1)$, so even if we have to rebalance every node on the path to the root, it’s still only $h \cdot O(1)$ rebalances.
Runtime in AVL Trees

- As usual, runtime of search, insert, and remove are all $O(\text{height})$

- How many nodes in an AVL tree of height $h$?

- or, what’s the tallest tree you can get with $n$ nodes?

  - Exact proof involves fibonacci sequence(!)

  - The answer is $O(\log n)$

- Intuition: To add to root’s height, you have to add to height of every subtree in one of root’s subtrees.