CS180 - AVL Trees (3)

Announcements
- Test in 1 week
- HW due Monday (download new BinaryTree.h)
- AVL tree will be done today

Today: delete
Removing in AVL trees

Step 1: Remove - just like in BST

Step 2: Re-balance (if removal violated H-B property.)

Note: Unlike insert, remove could actually unbalance all the way to the root.
Example:

remove(44)

remove(32) (pivot x up twice)
Fixing the tree
Algorithm to remove

- Remove as in BST
- Track lower node that was removed.
- Travel up tree, searching for unbalanced nodes (fixing) until you reach the root.
Performance

For insert & delete follow root to leaf path at most 3 times:
- find
- next in inorder (for remove)
- travel back up tree balancing

At each node:
- 1 comparison
- 2 get higher child, 1 set height
  2 lift, + 2 pivot
  full pointer updates

Total time: $O(lgn)$
pivot(y)

reset z's height then y's
Testing insert:

[10, 5, 15] → [21, 35, 42]