CS180 - AVL trees (2)

Note Title 11/21/2011

Announcements

- No office hours today.
  (Tomorrow?)
- Checkpoint today.
- Program due Sunday.
- Redo on 1 problem from midterm - due next Monday.
- HW4 - due Wed., Nov. 30.

AVL trees

Main property:

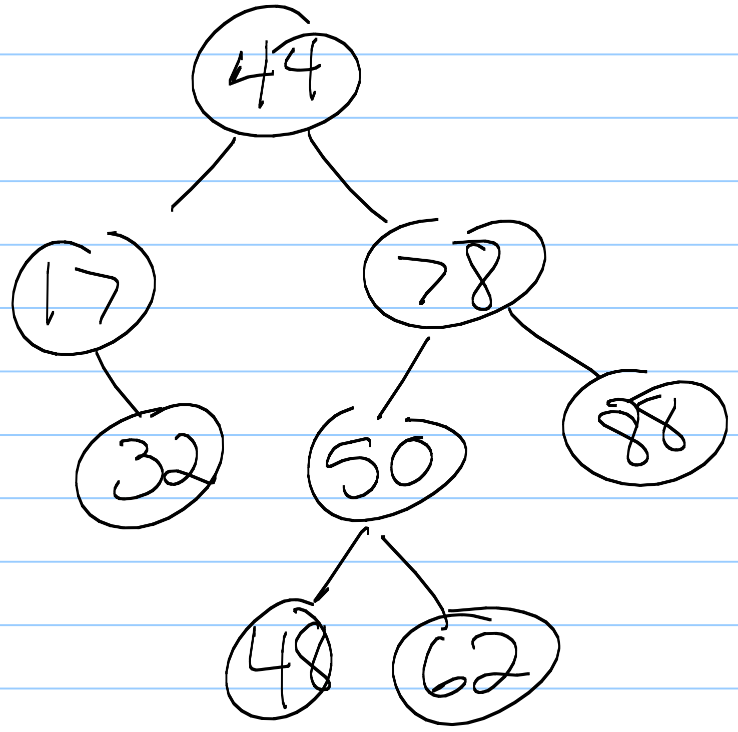
How to mess up balancing?

So: consider the lowest node which does not satisfy height-balance property - call this z.

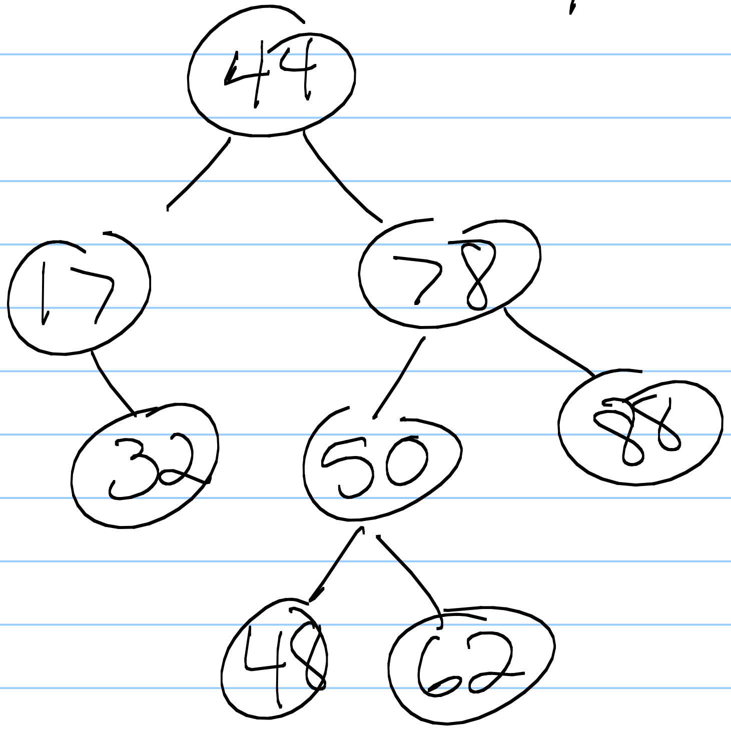
Let y be z's child with larger height.

Let x be y's child with larger height.

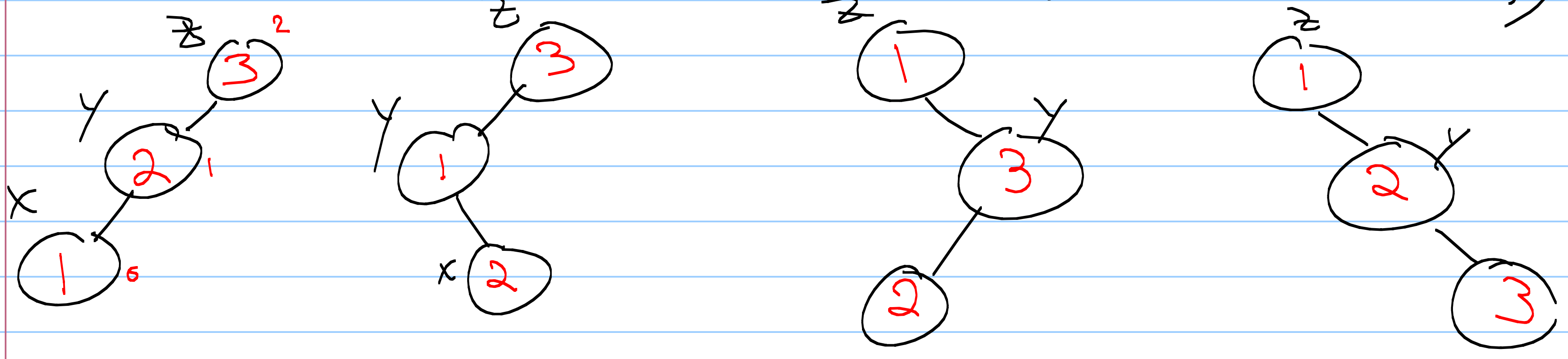
Now - fix it!



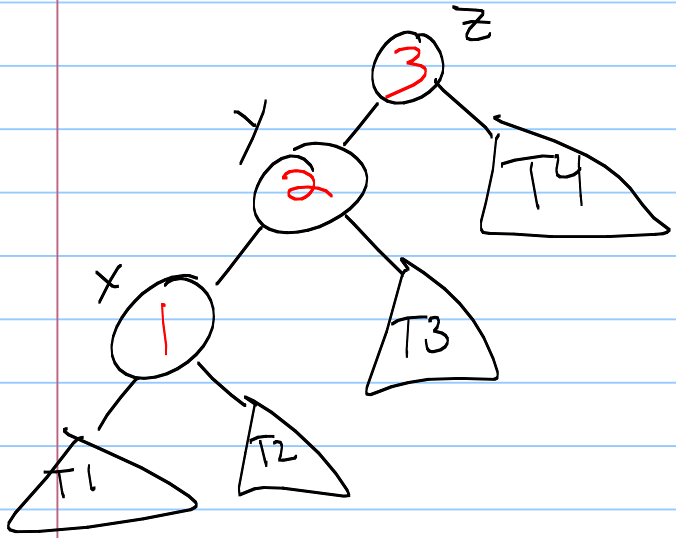
What did you do?



Generalize – Consider x, y, & z. How can we restructure?
(Hint: What is inorder traversal of these in each case?)

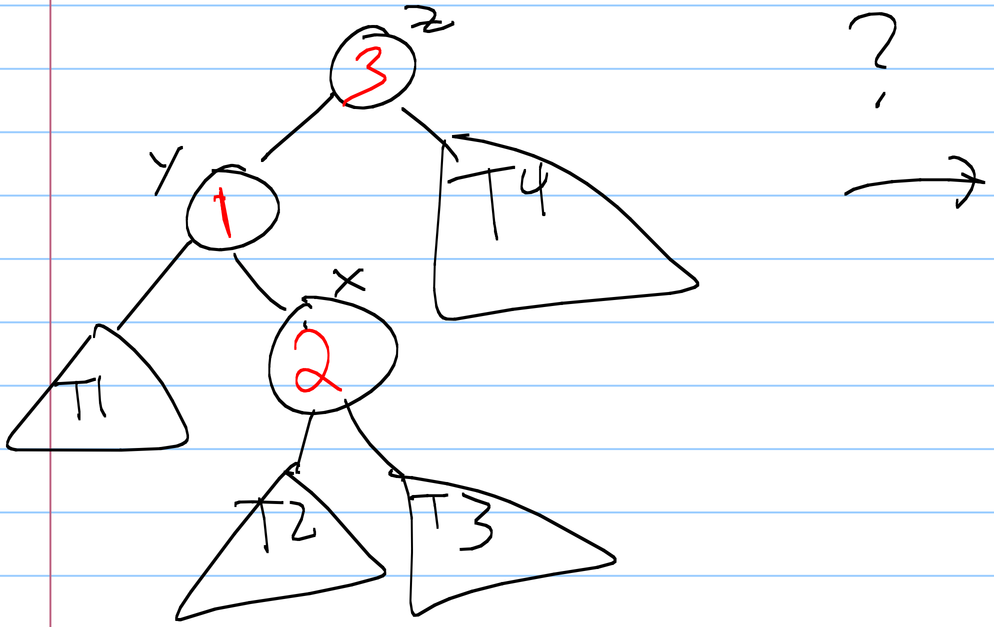


Actual picture:



Where do the subtrees go??

Another



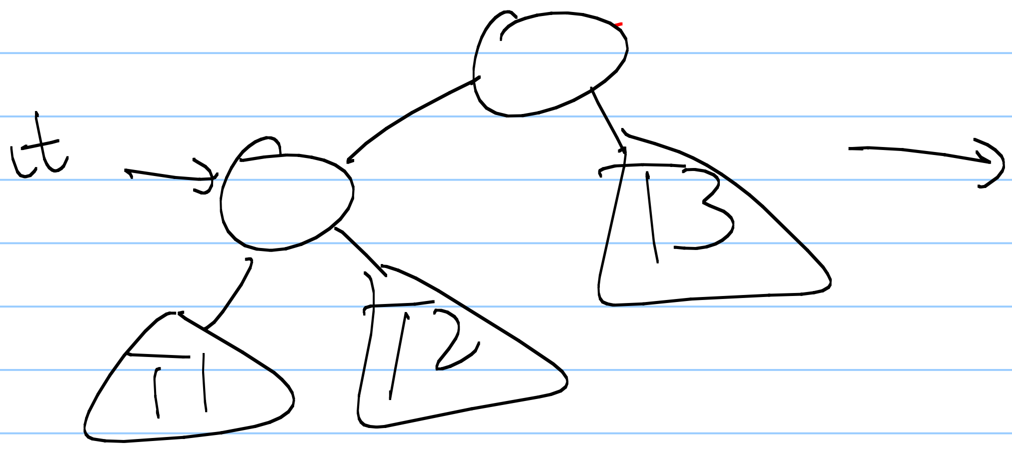
Any way you do this, "2" becomes the root of the new subtree, with "1" to the left & "3" to the right!

What about T1, T2, T3, & T4?

So how can we code this?

Back to BinaryTree.h:

_pivot(it) will swap it and its parent



Removing in AVL trees

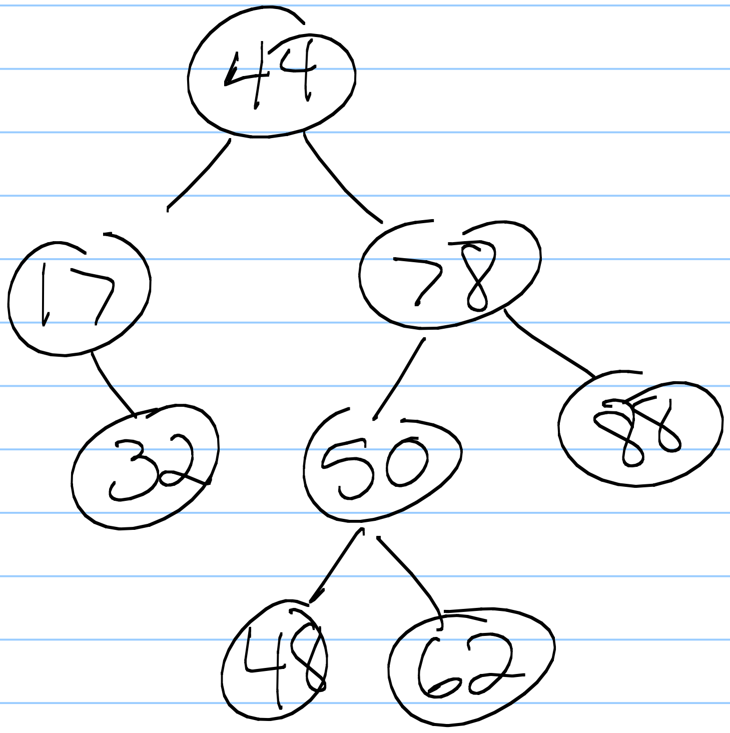
Step 1: Remove - just like in BST

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)



Then: remove (17)

Fixing the tree