

# 180 - AVL trees

Note Title

4/13/2011

## Announcements

- Checkpoint on Friday
- Final is Wed at noon - 5/11  
Check if you have a conflict!  
email me by next Monday!

# Balanced Binary Search Trees

Runtimes for BSTs:  $O(n)$

Why is this bad?

Lists: find  $O(n)$  insert  $O(1)$

Vectors:  $O(1) - O(\log n)$   
 $O(n)$   $O(n)$

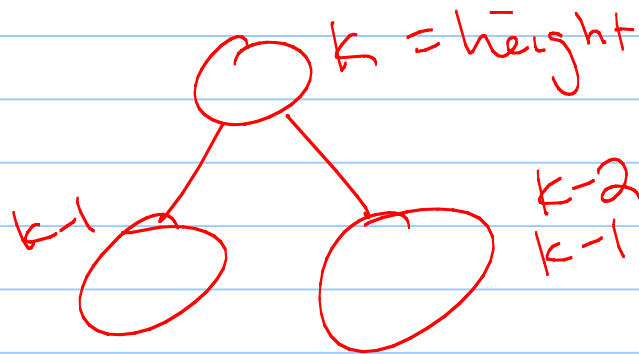
Goal: Do better, but  $O(n)$  is worse.

# AVL Trees:

Height - Balance Property:

For every node of  $T$ , the heights of the children differ by at most 1.

$$\Rightarrow \text{max height} \leq 2 \lceil \log_2 n \rceil$$



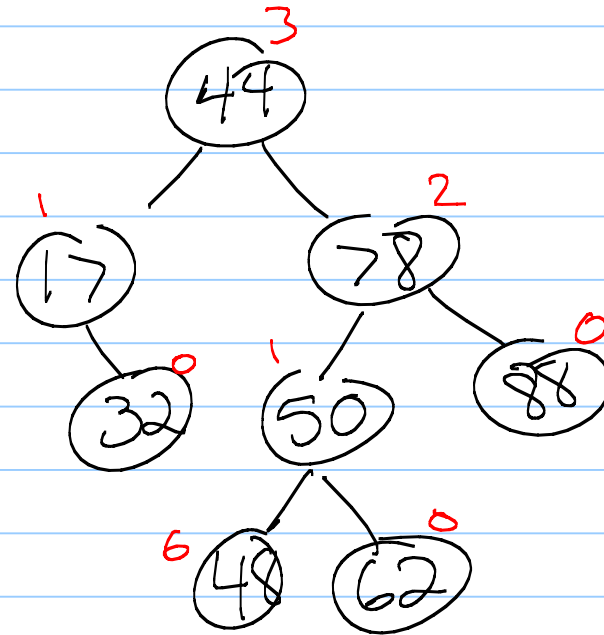
Aside:  
Red-Black Trees  
 $\approx 1.6 \dots \log_2 n$

Now: How can we mess this up?

(In other words,  
how can the  
height change?)

insert

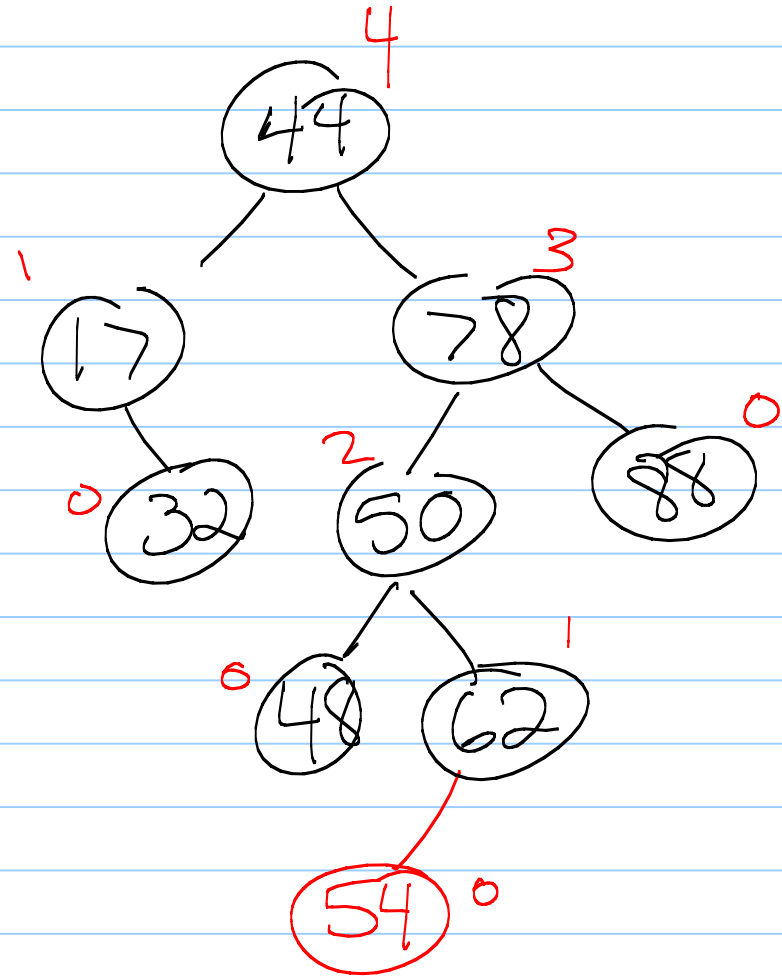
remove



Insert:

insert(54)

Bad - need  
to fix

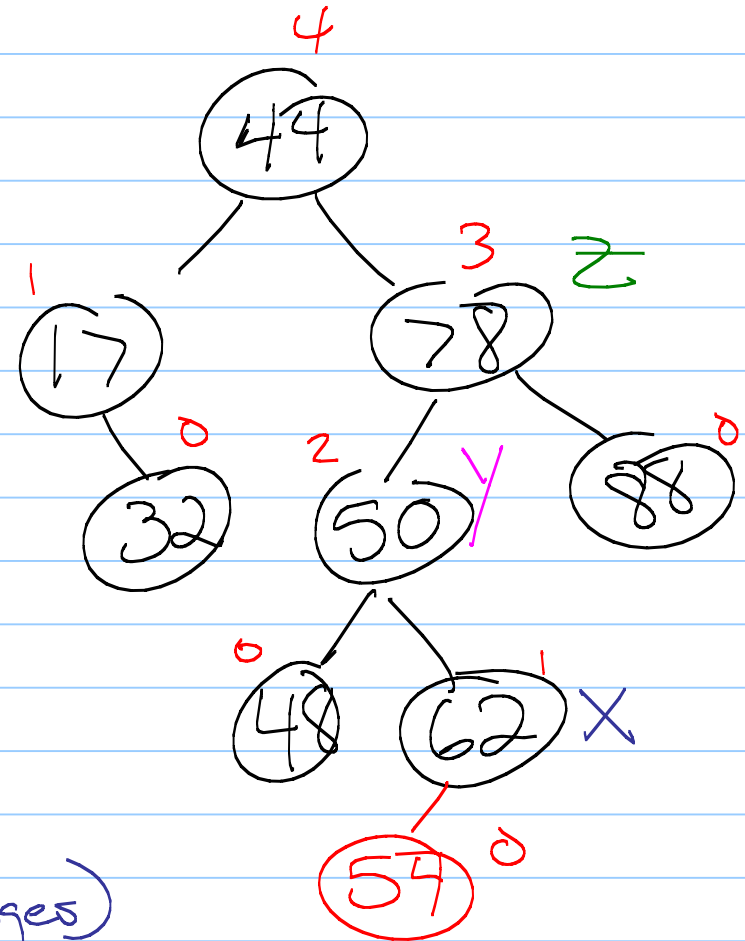


So: consider the lowest node which does not satisfy height-balance property - call this  $z$ .

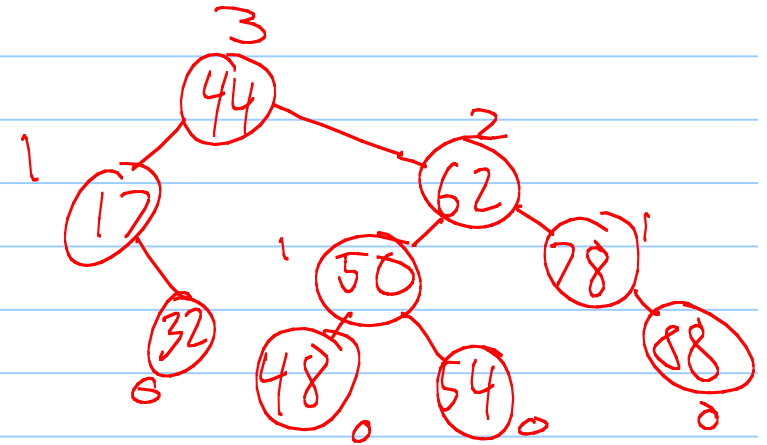
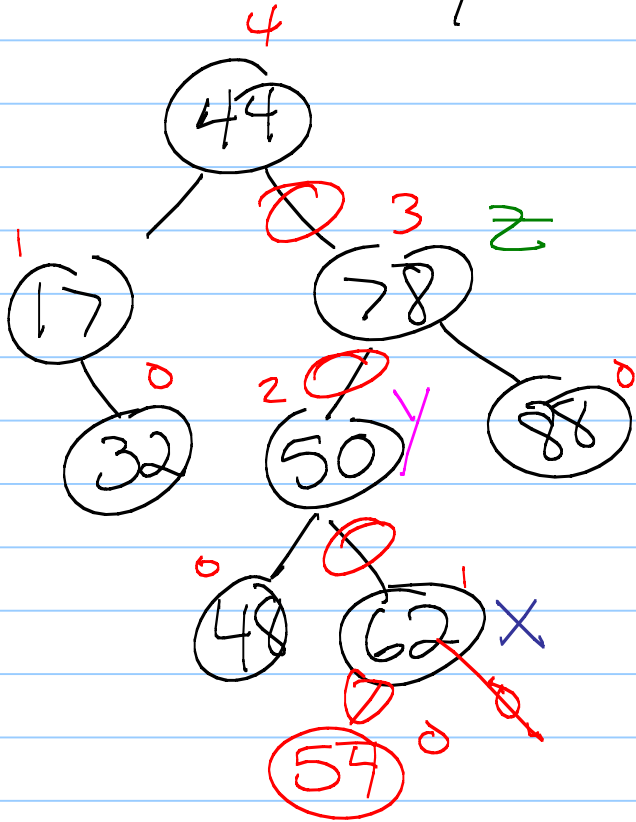
Let  $y$  be  $z$ 's child with larger height.

Let  $x$  be  $y$ 's child with larger height.

Now - fix it! (goal: use few changes)



What did you do?



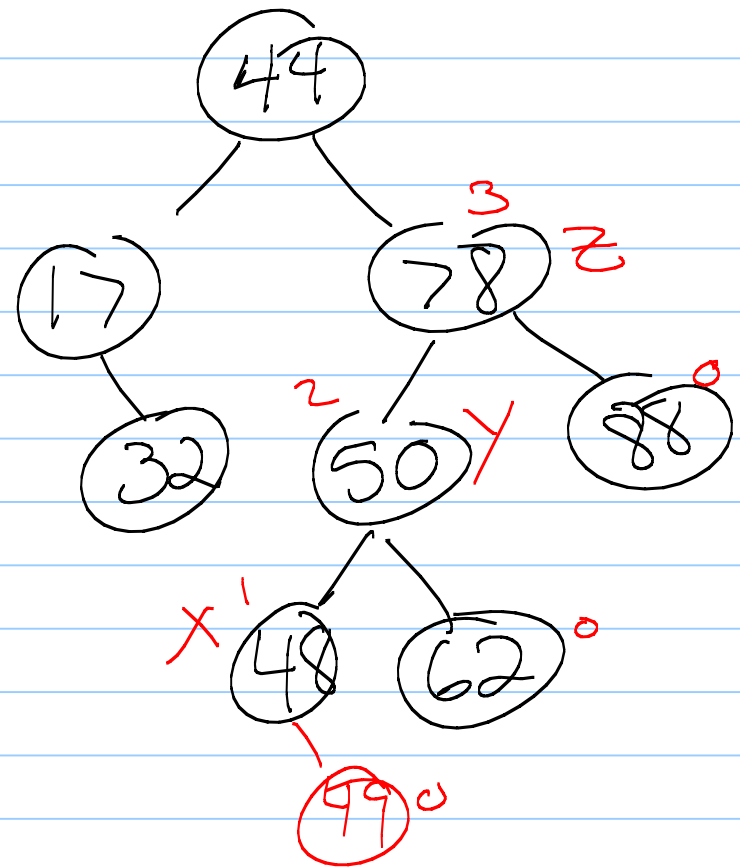
~ 10 or 12 pointer changes

Another - insert (49)  
So: consider the lowest node which does not satisfy height-balance property - call this  $z$

Let  $y$  be  $z$ 's child with larger height.

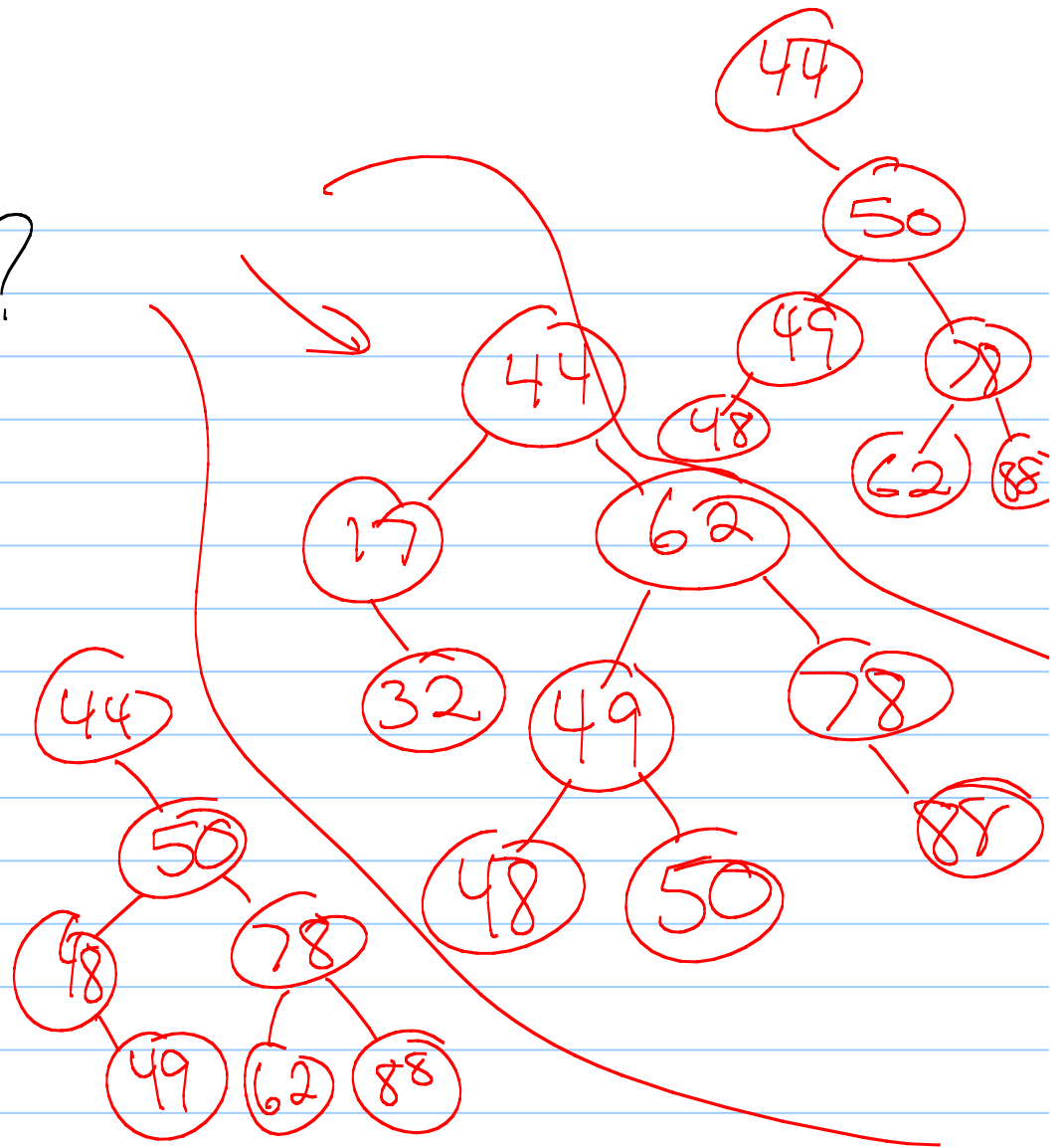
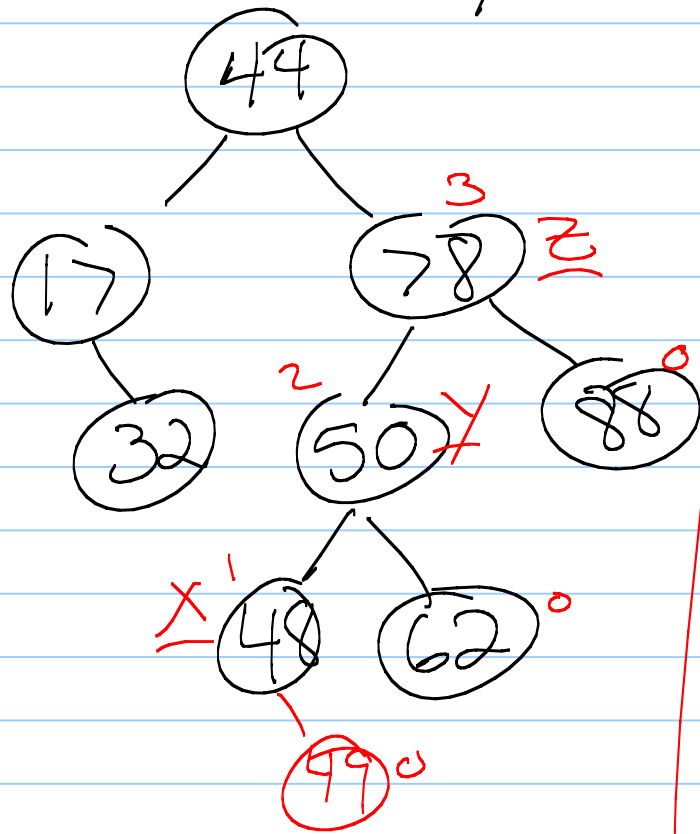
Let  $x$  be  $y$ 's child with larger height.

Now - fix it!



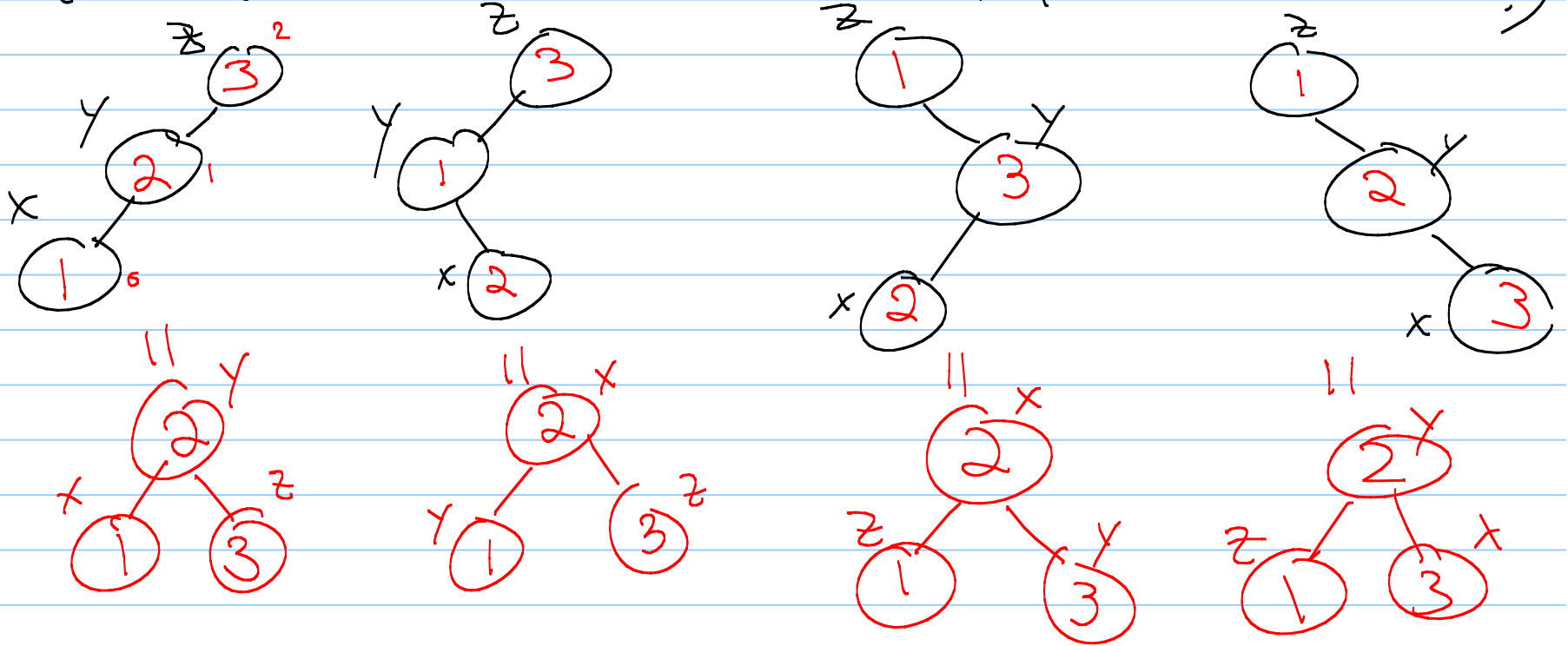


What did you do?

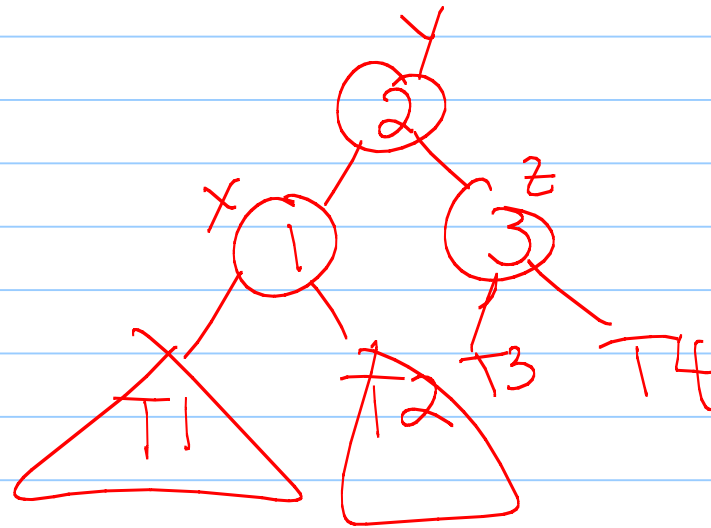
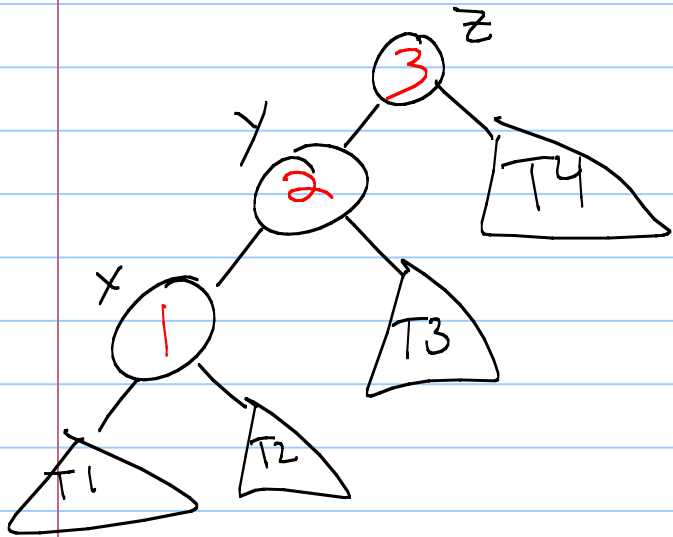


Generalize - Consider  $x, y, z$ . How can we restructure?

(Hint: What is inorder traversal of these in each case?)

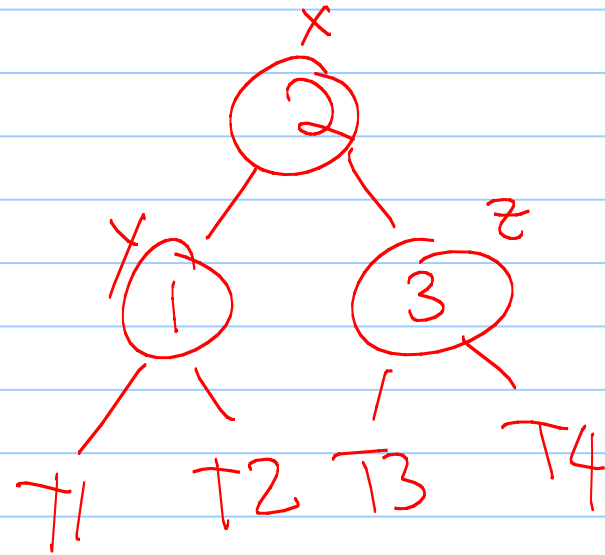
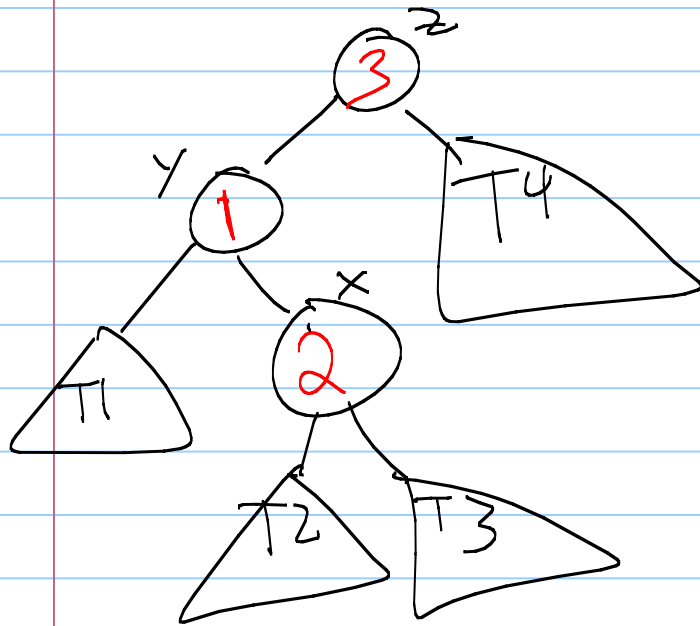


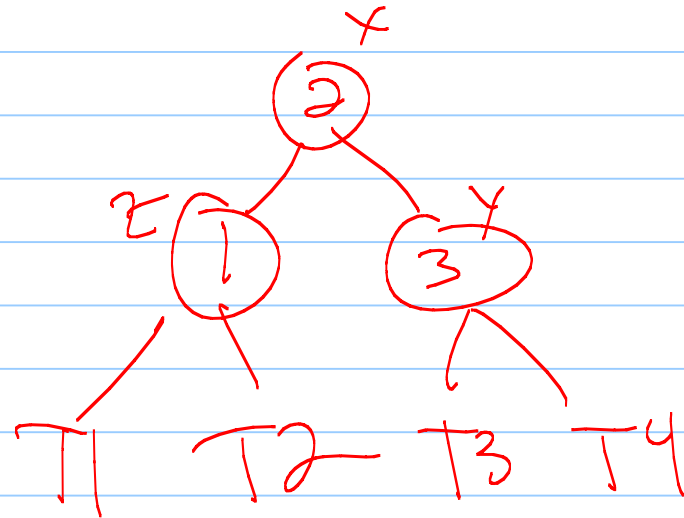
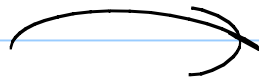
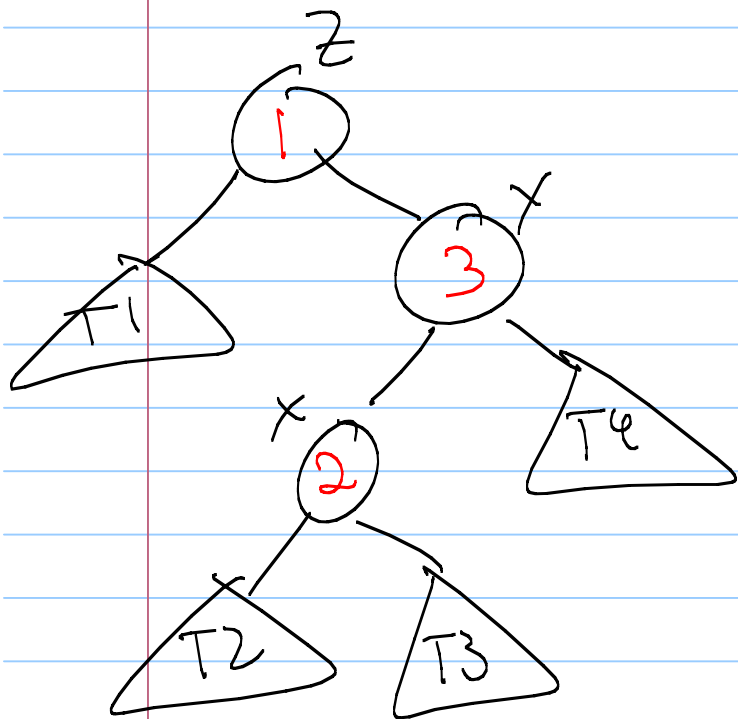
Actual picture:



Where do the subtrees go??

Another





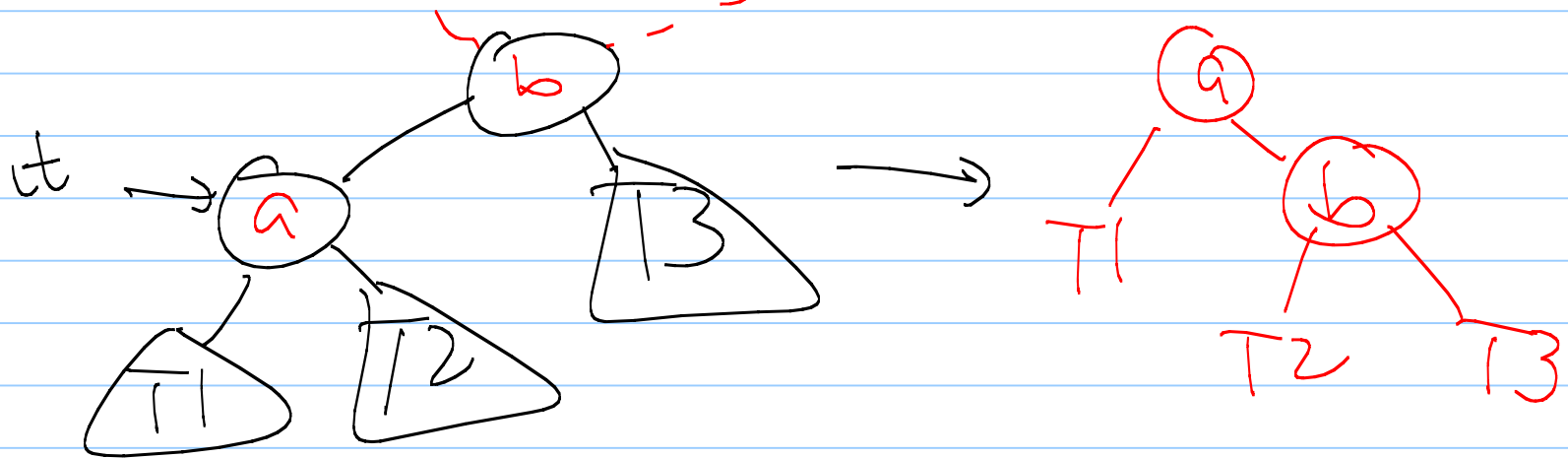
Any way you do this, "2" becomes  
the root of the new subtree,  
with "1" to the left & "3" to  
the right!

What about T1, T2, T3, & T4?

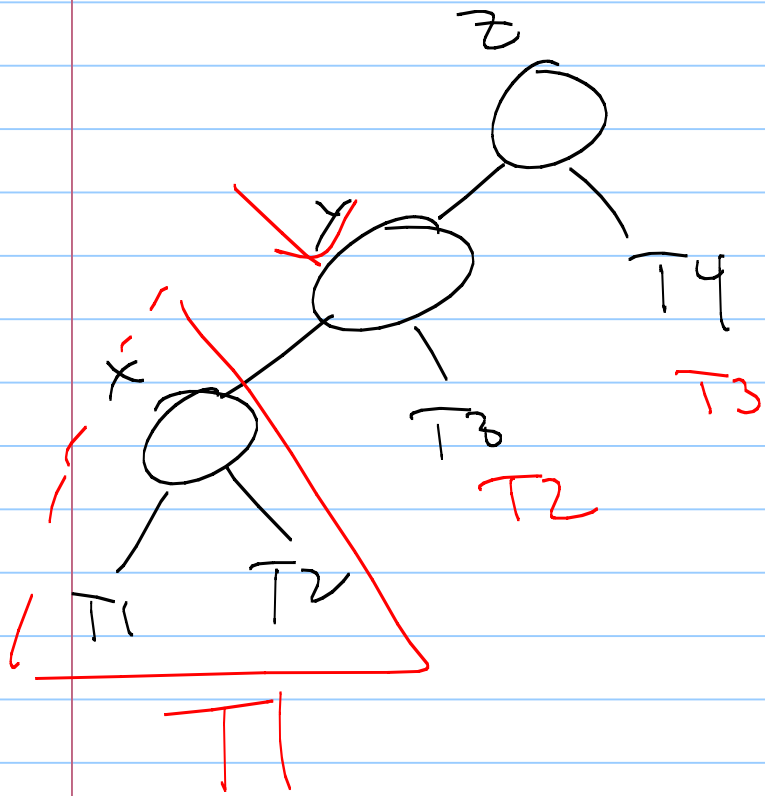
So how can we code this?

Back to Binary Tree.h:

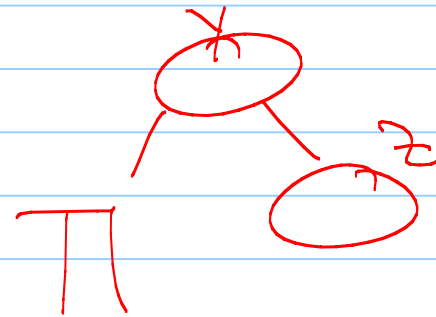
- pivot (it) will swap it  
and its parent



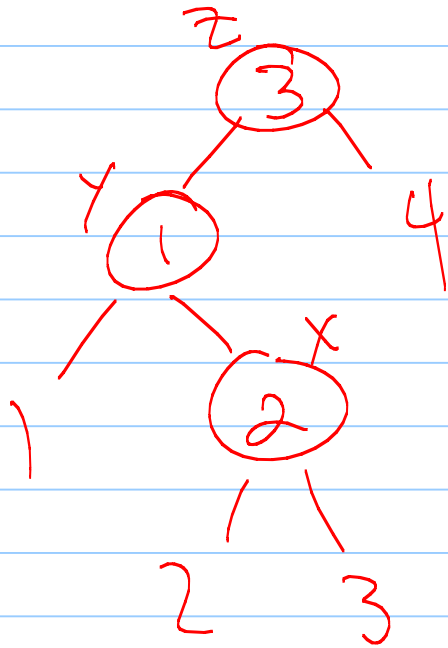
how to use -pivot? (who is new root?)



-pivot( $y$ )







- pivot(x)  
- pivot(x)