

CS180

Note Title

11/18/2011

Announcements

# Remove in a BST

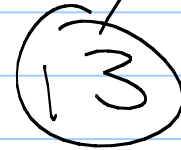
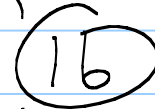
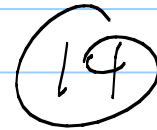
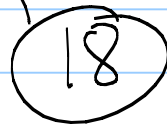
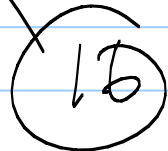
Several cases: Let  $v$  be our target node to delete



When is it easy?

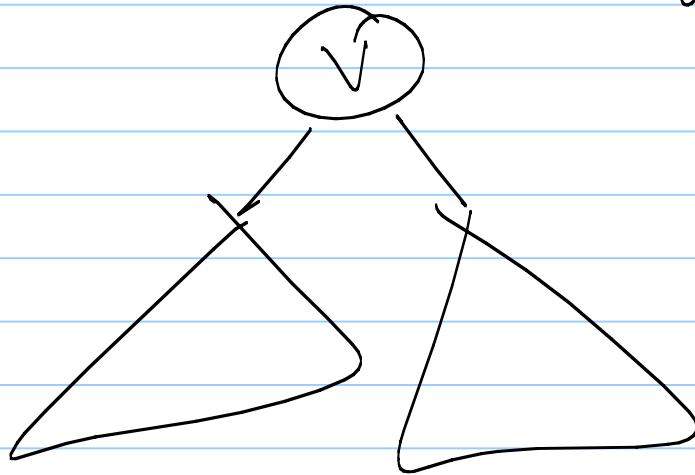
Case 1:  $v$  is a leaf or  $v$  has only 1 child.

Ex:

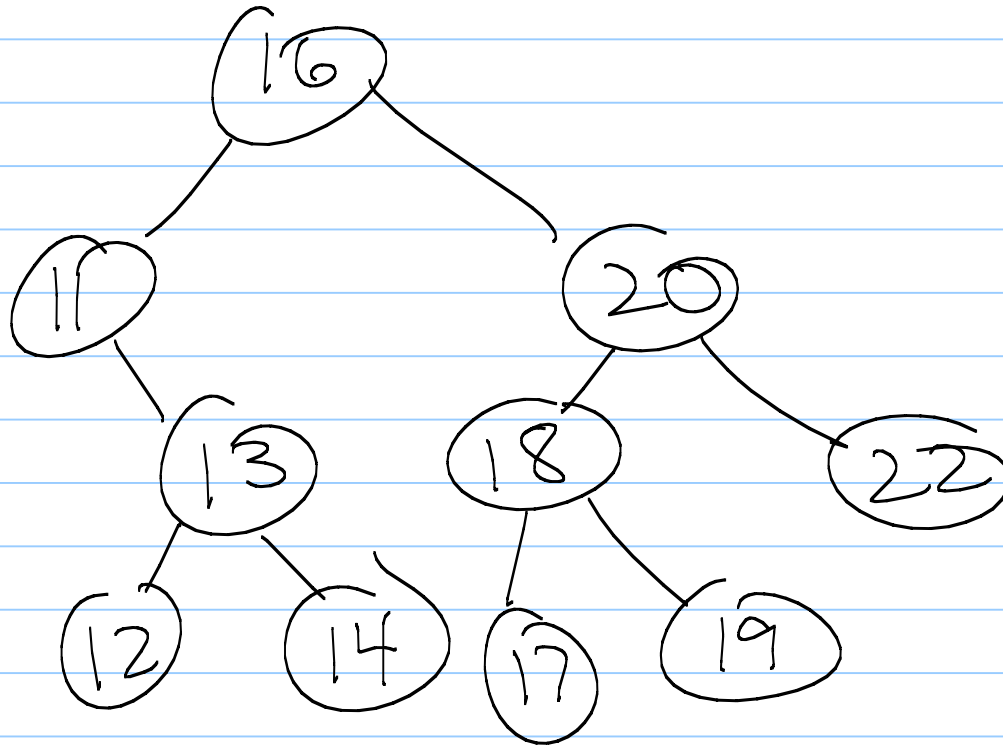


Case 2:  $v$  has two children

What can go in  
 $v$ 's spot?



Ex:

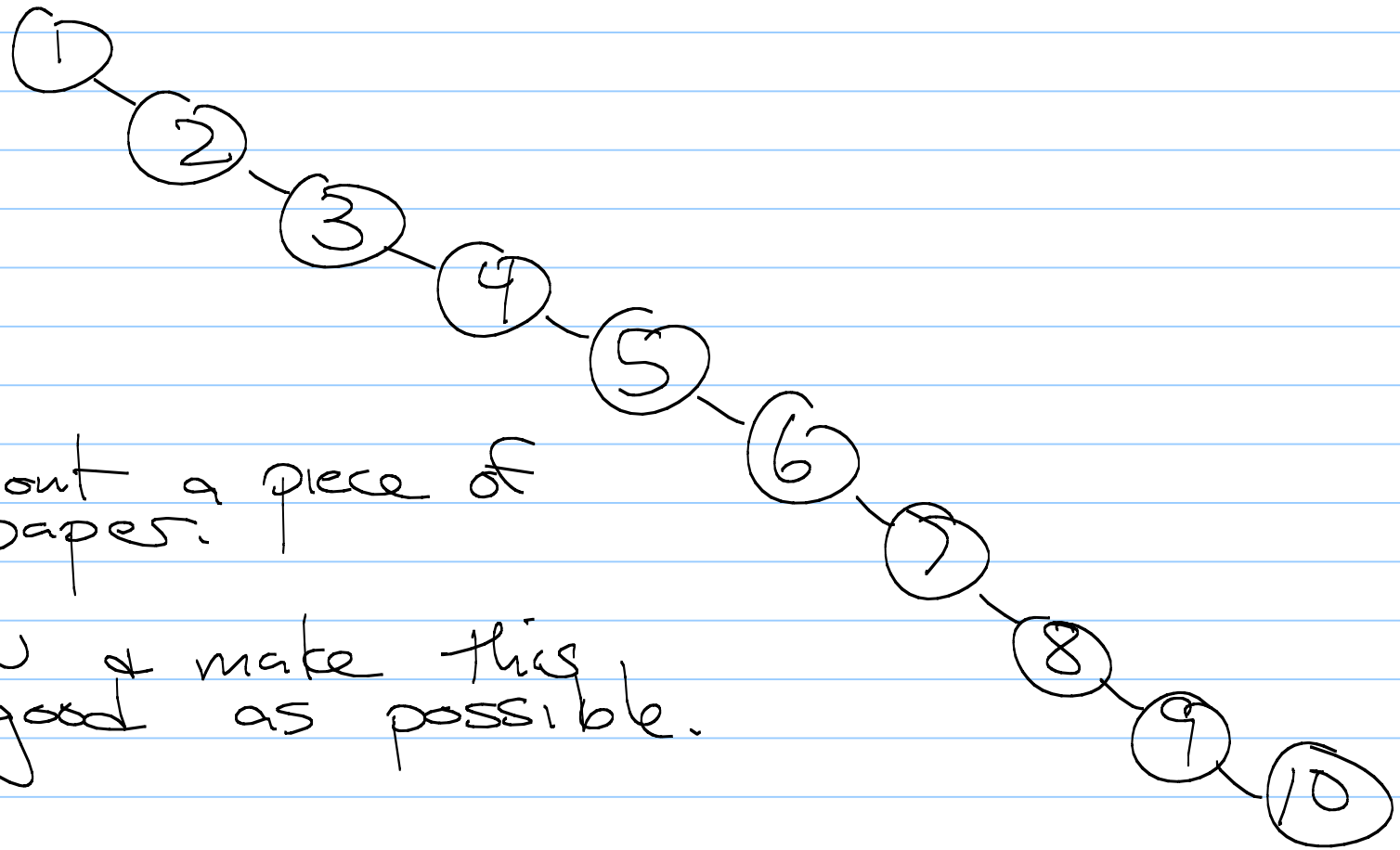


Key: Next node in an inorder traversal has valid value and can have at most one child.

Why? It can't have a left child.

(why?)

Consider this tree:



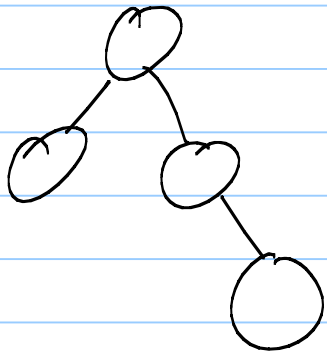
Take out a piece of paper.

Redraw & make this as good as possible.

# AVL Trees

Height - Balance Property:  
For every node of  $T$ , the heights of the children differ by at most 1.

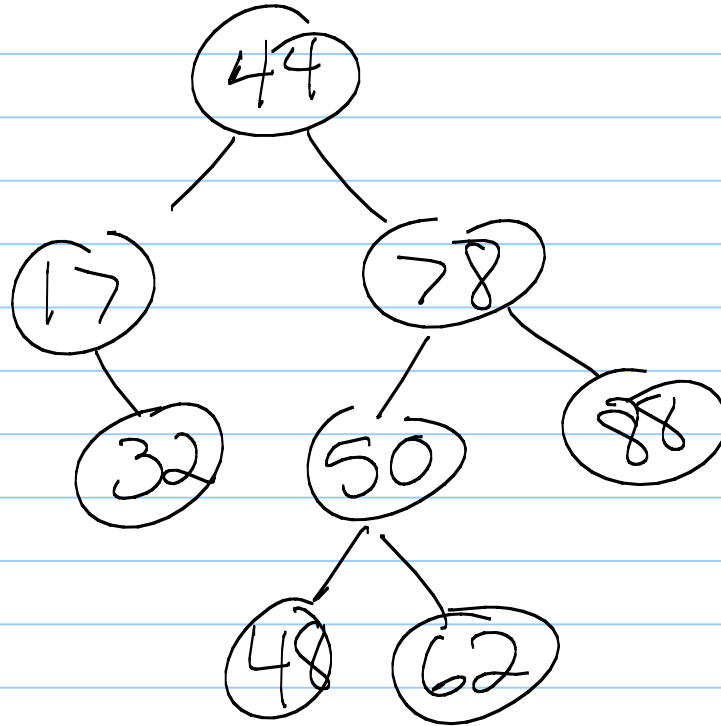
$\Rightarrow$  max height  $\leq$



(How do we calculate height again?)

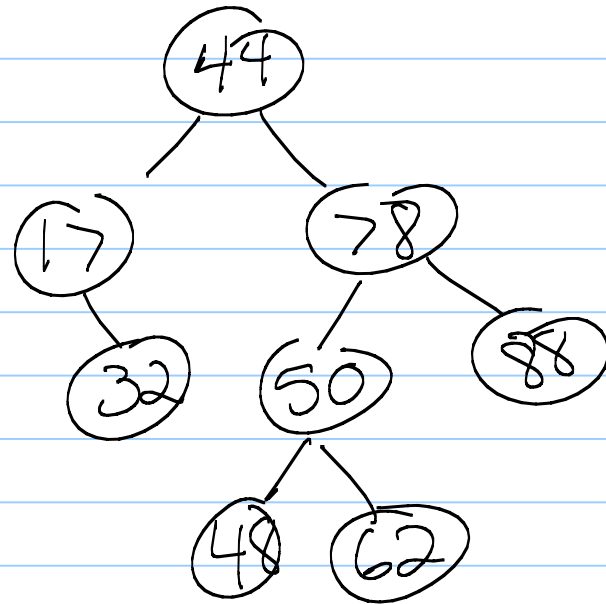


Ex:



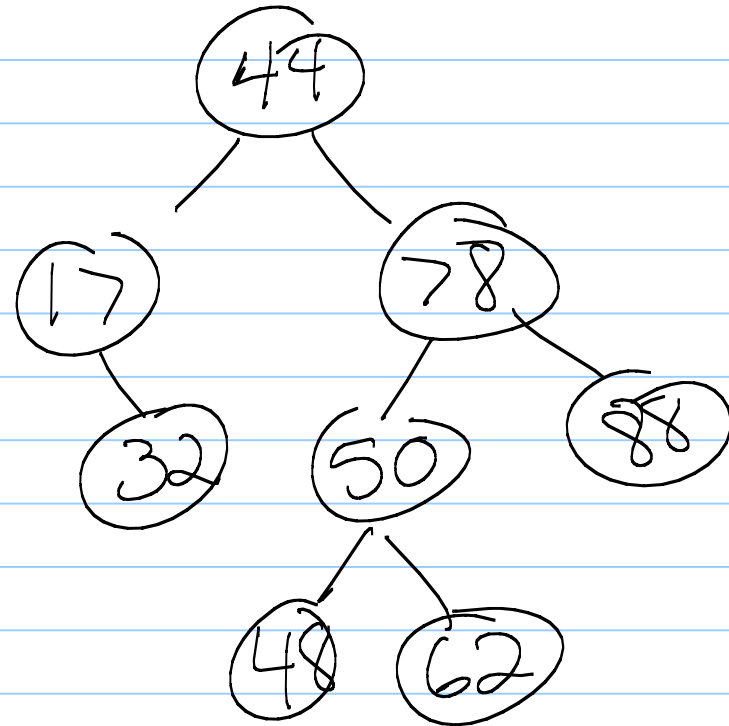
Now: How can we mess this up?

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



Insert:

insert(54)

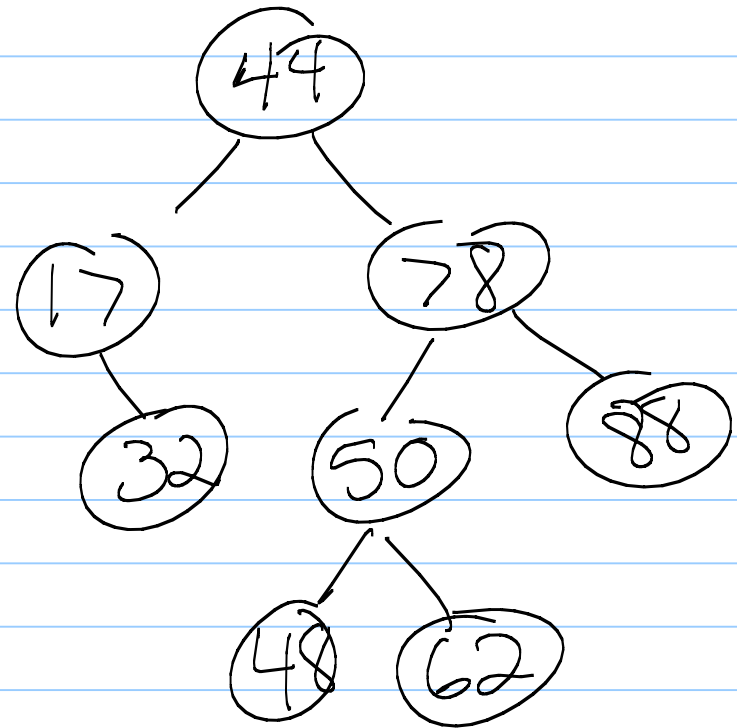


So: consider the lowest node which does not satisfy height-balance property - call this  $z$ .

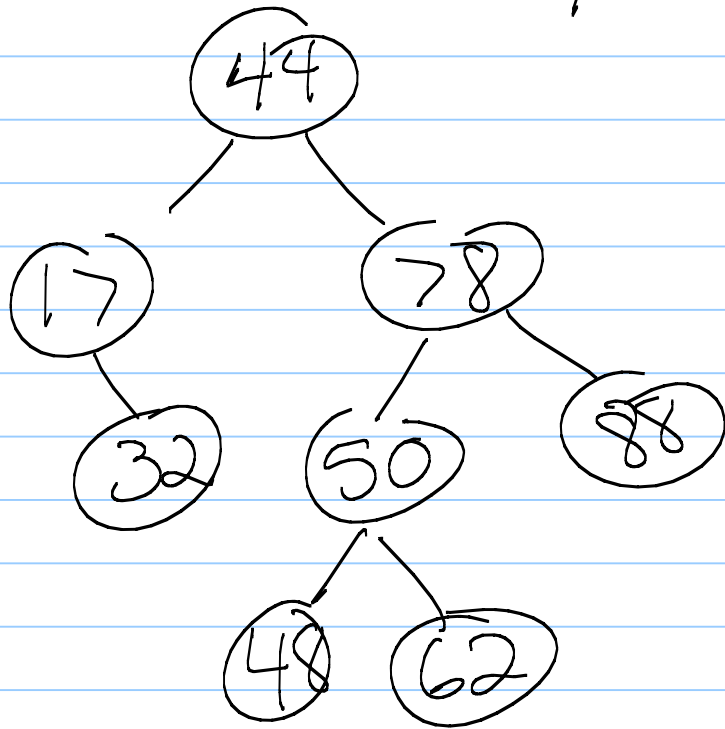
Let  $y$  be  $z$ 's child with larger height.

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

Now - fix it!



What did you do?



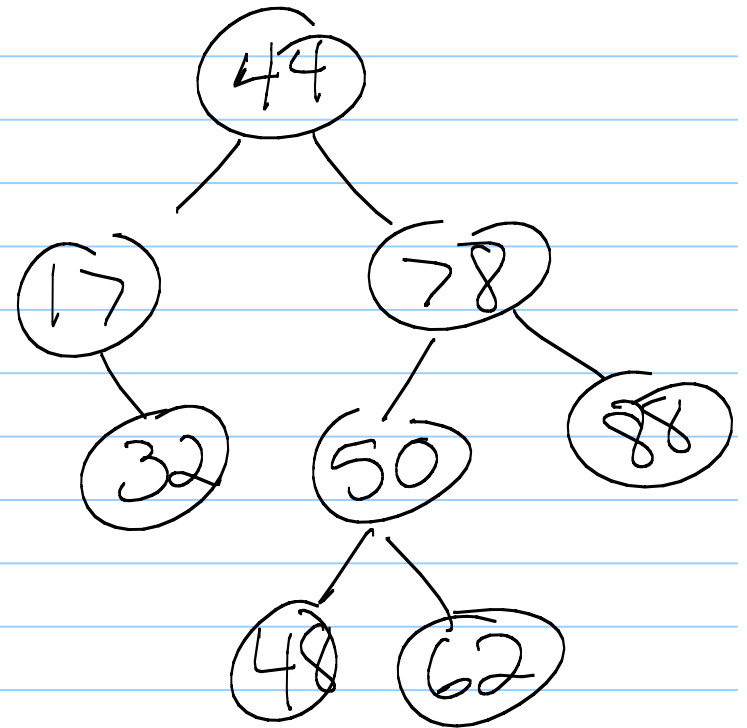
Another - insert (49)

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

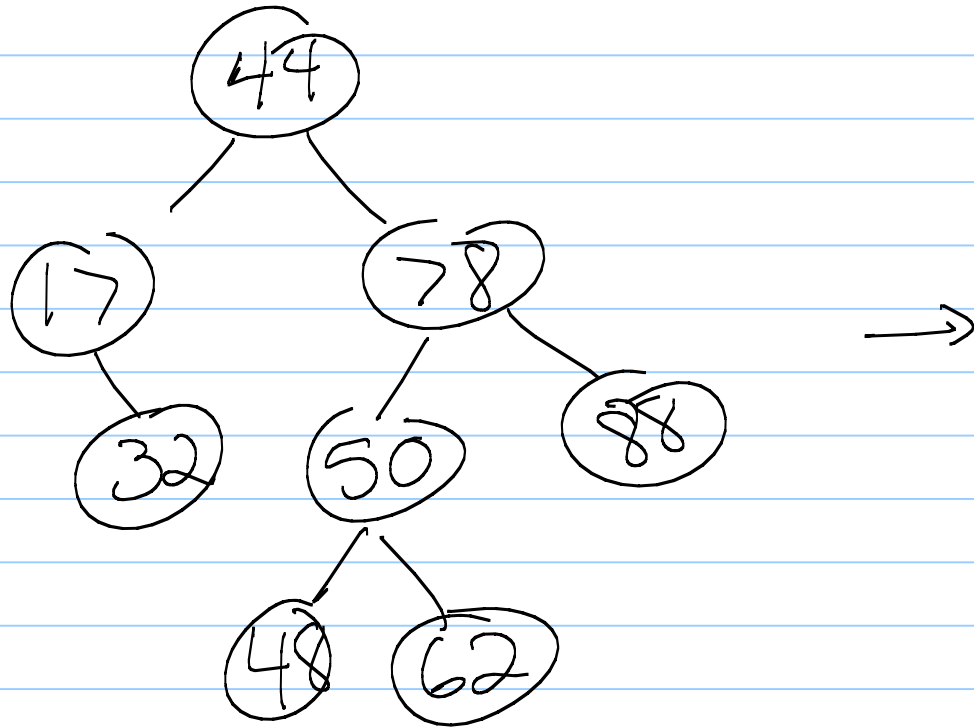
Let  $z$  be  $x$ 's child with larger height.

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

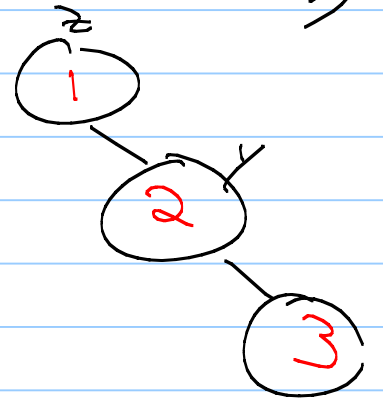
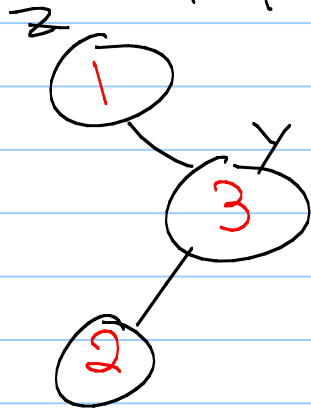
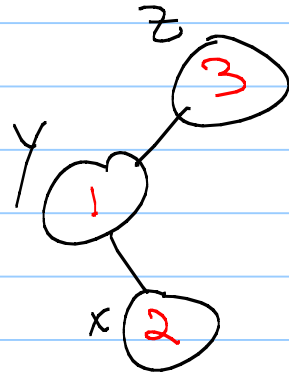
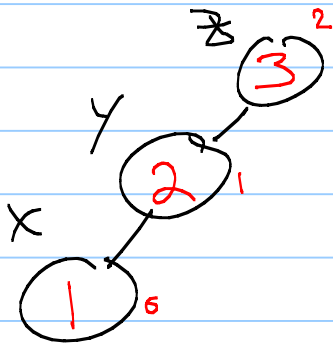
Now - fix it!



What did you do?

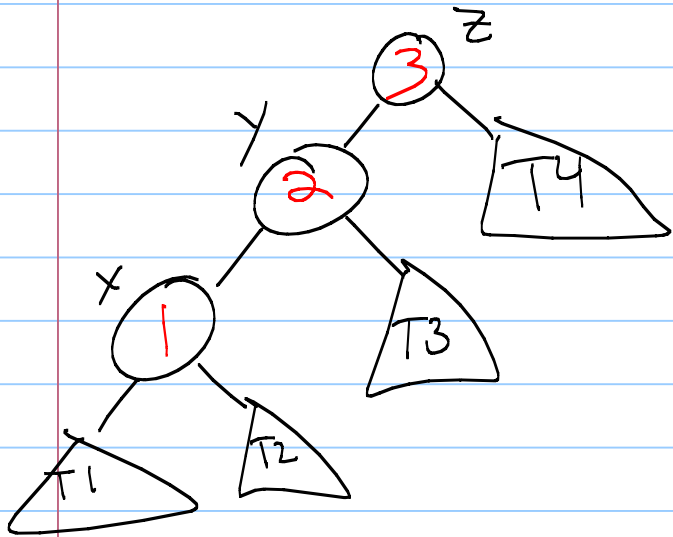


Generalize - Consider  $x, y, \& z$ . How can we restructure?  
(Hint: What is inorder traversal of these in each case?)



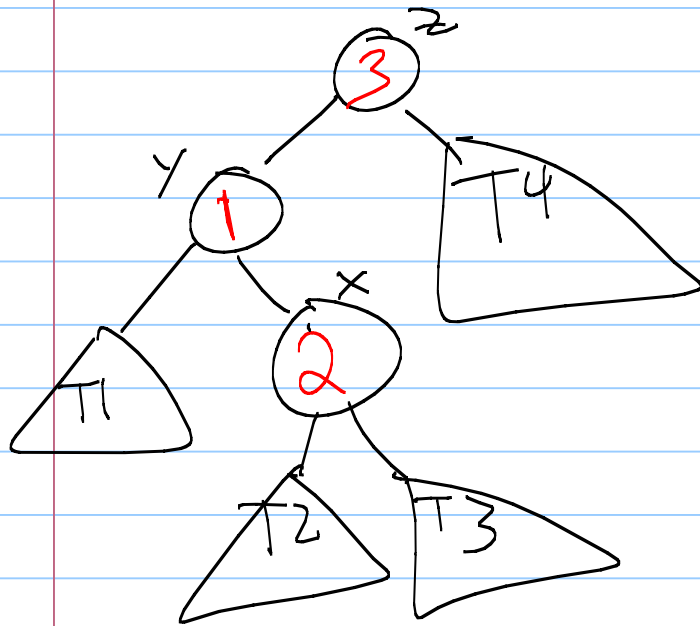


Actual picture:



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?

