CS 180 - AVL Trees

Note Title 10/19/2012

Announcements

- HW due now
- Next HW - remove in a BST

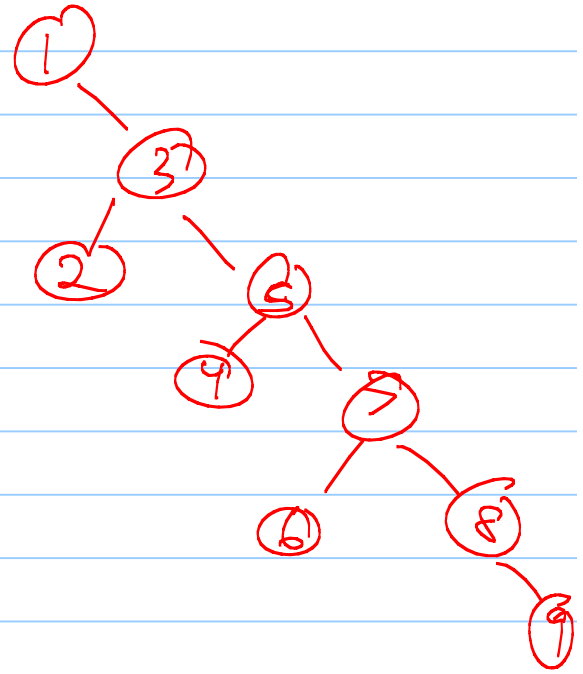
Recap: BST

Runtimes:

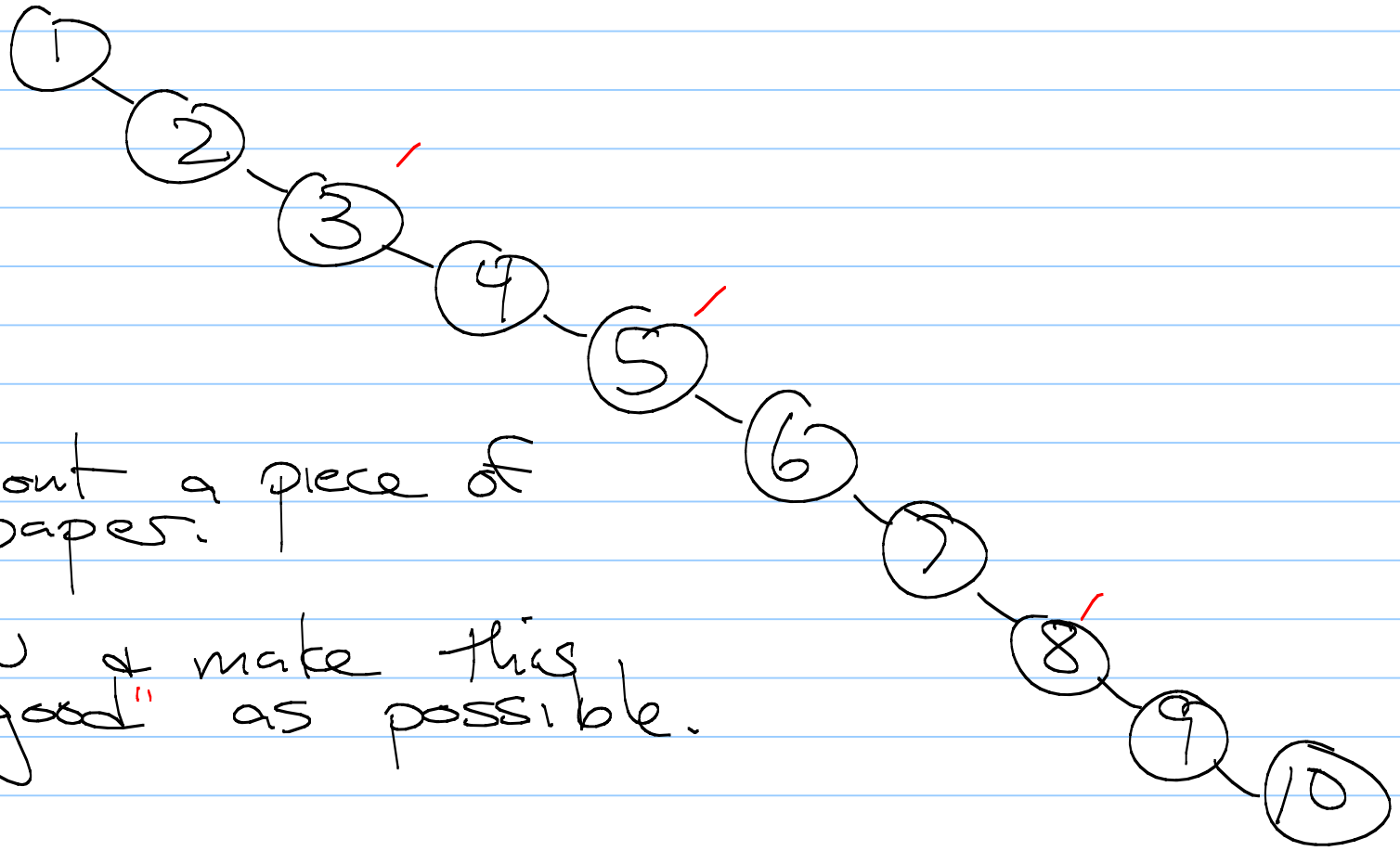
Insert: $O(n)$

Find: $O(n)$

Remove: $O(n)$



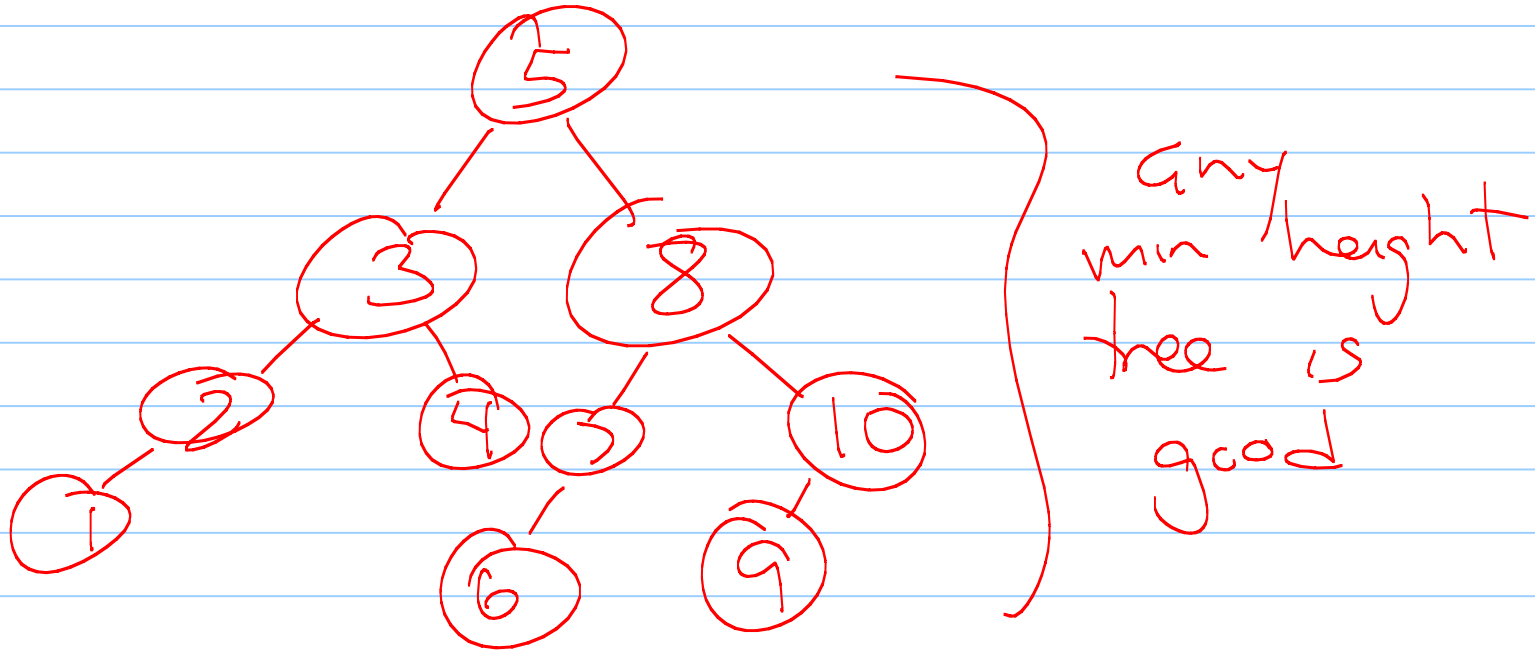
Consider this tree:



Take out a piece of paper.

Redraw & make this as "good" as possible.

What did you do?



Balanced Binary Search Tree

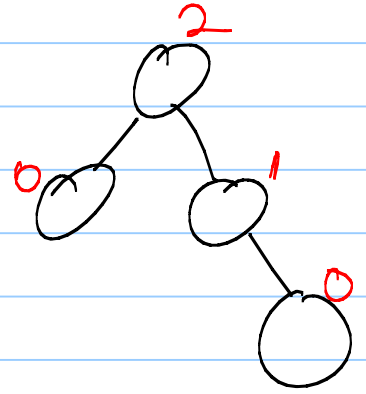
- Red-black trees
- Splay Trees
- AVL trees ←
- ⋮

Goal of all: "balance" the tree
O(log n)

AVL Trees : BST with:

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

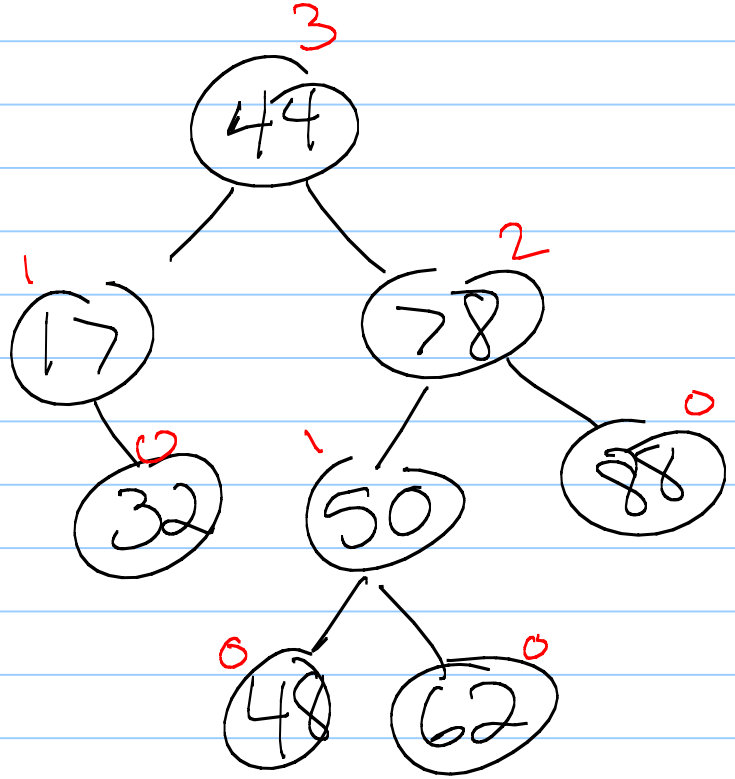
$\Rightarrow$ max height $\leq 2 \cdot \lceil \log_2 n \rceil$



(How do we calculate height again?)

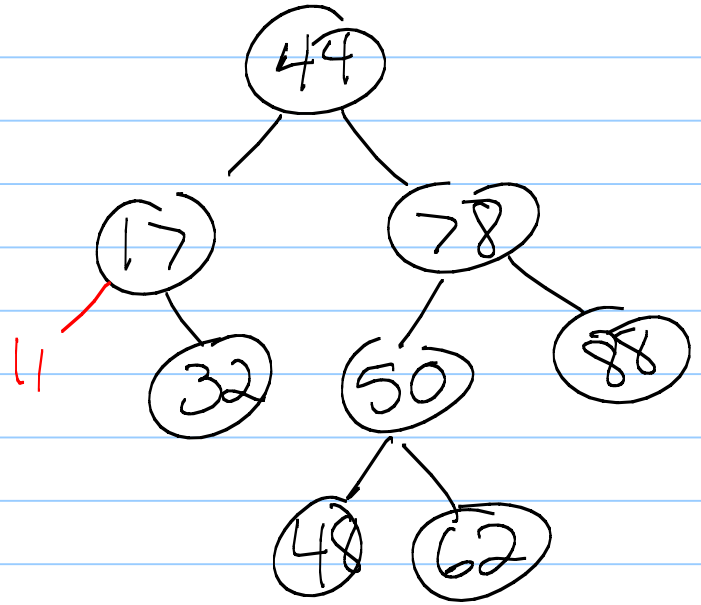
$h(v) = \max(h(\text{left}(v)), h(\text{right}(v)) + 1$

Ex:



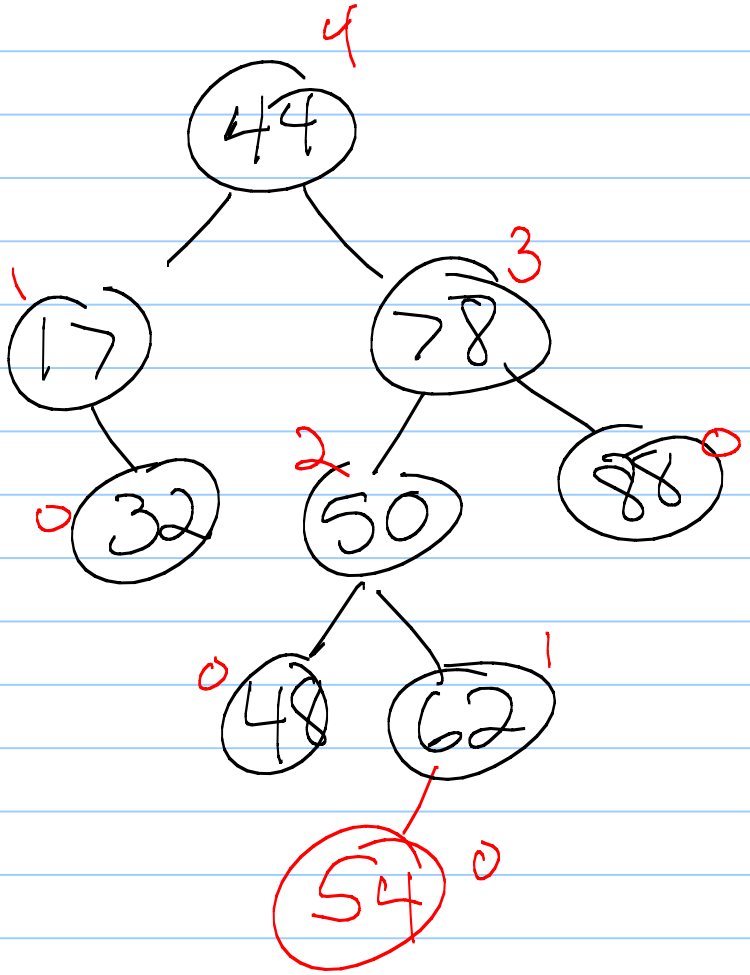
Now: How can we mess this up?

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



Insert:

insert(54)

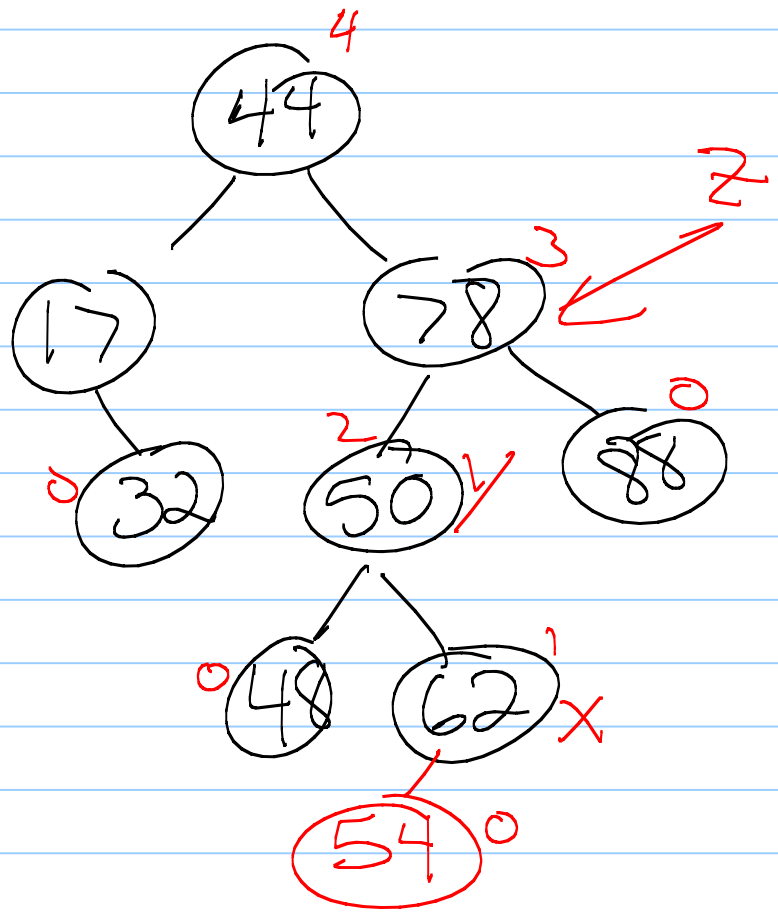


So: consider the lowest node which does not satisfy height-balance property -- call this z.

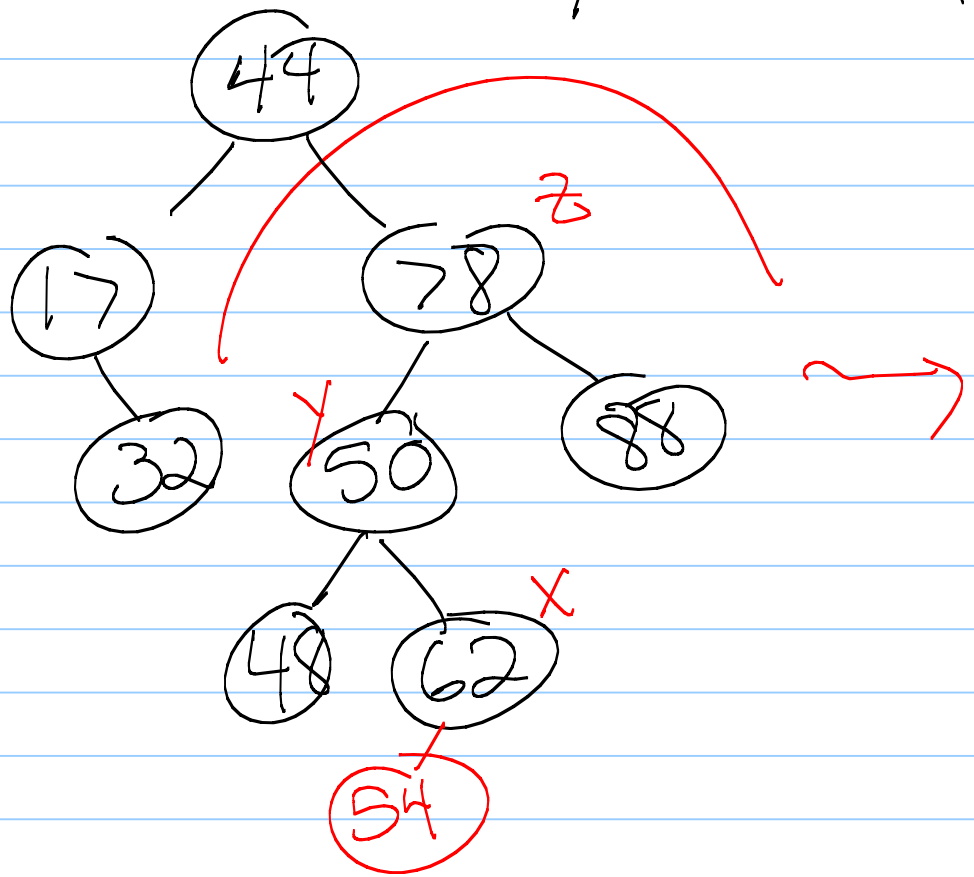
Let y be z's child with larger height.

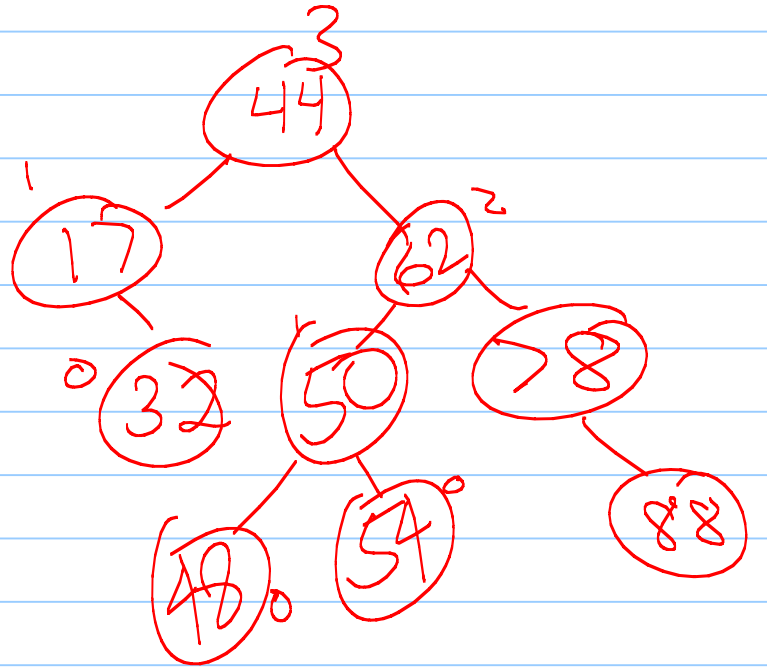
Let x be y's child with larger height.

Now - fix it!



What did you do?



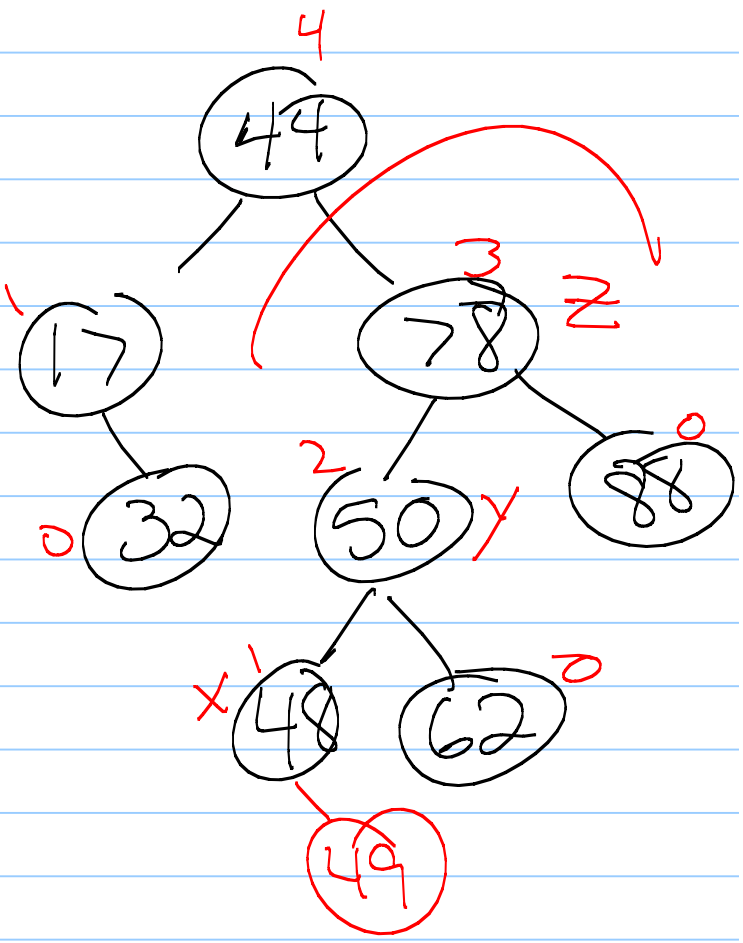


Another - insert (49)
So: consider the lowest node which does not satisfy height-balance property - call this z.

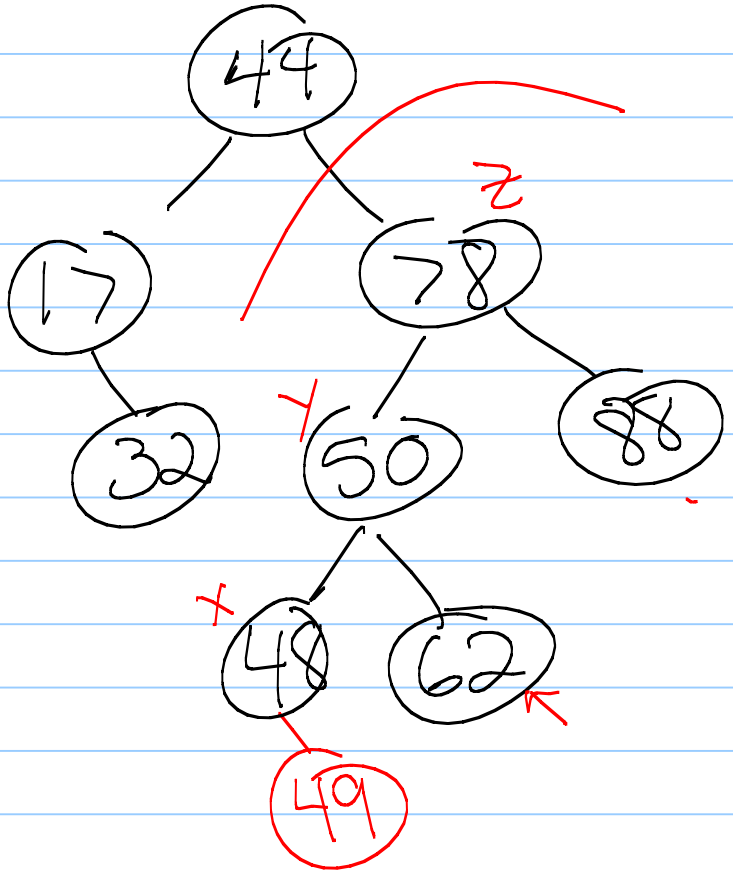
Let y be z's child with larger height.

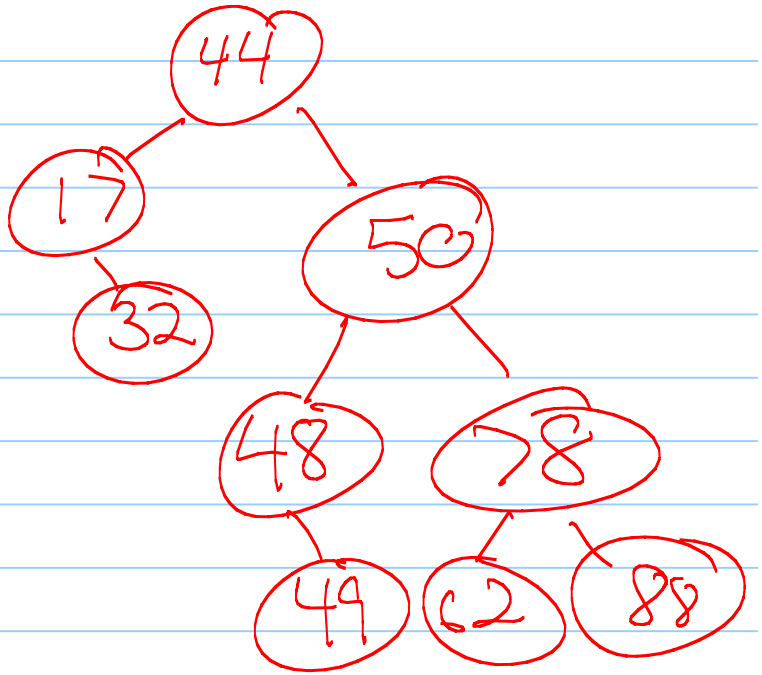
Let x be y's child with larger height.

Now - fix it!



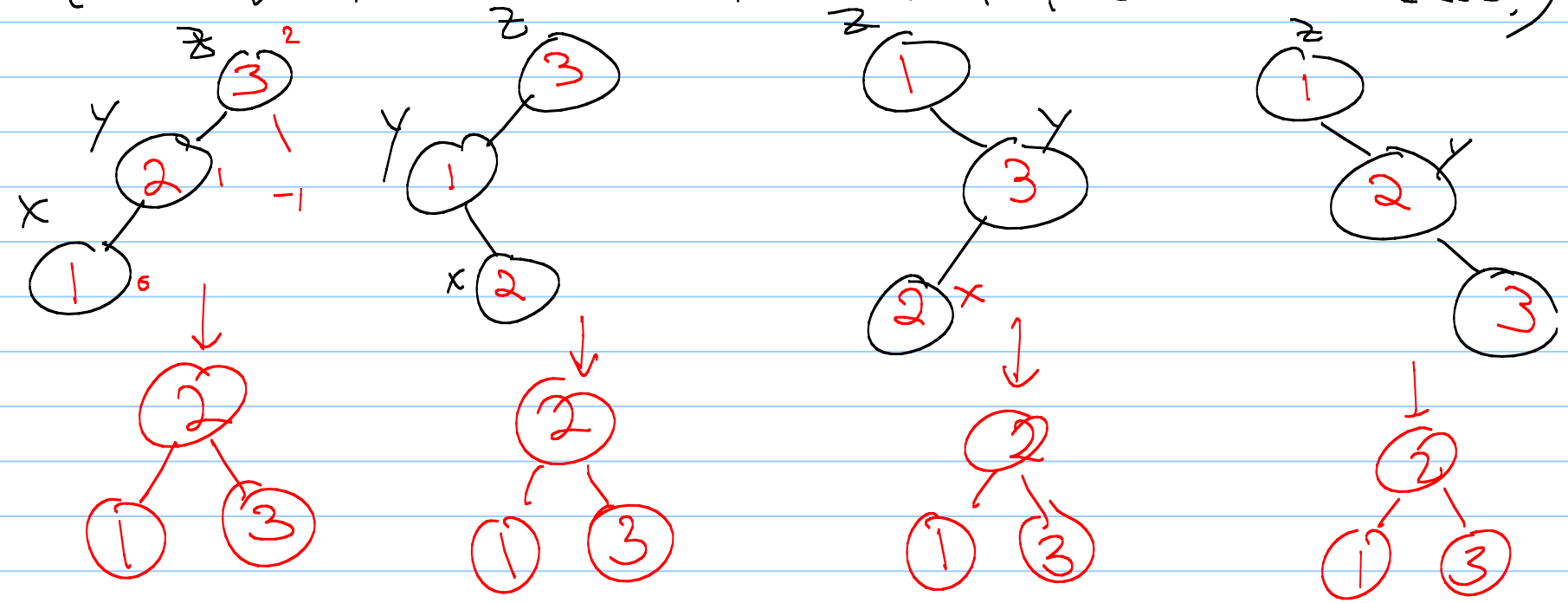
What did you do?



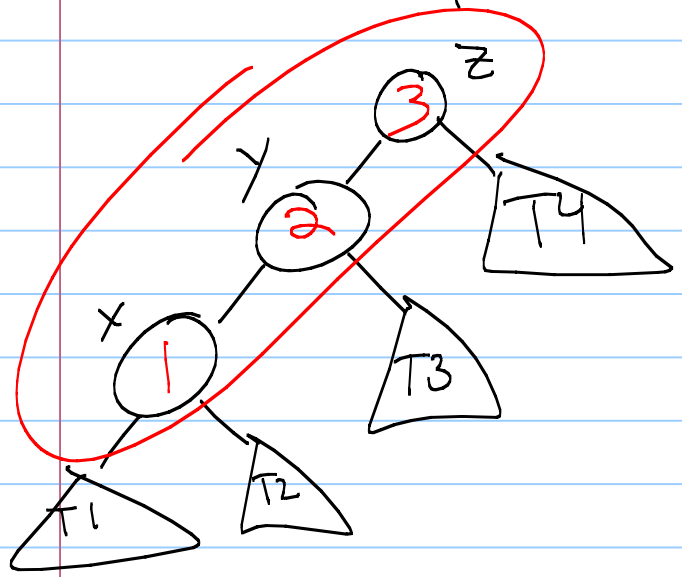


Generalize – Consider x, y, & z. How can we restructure?

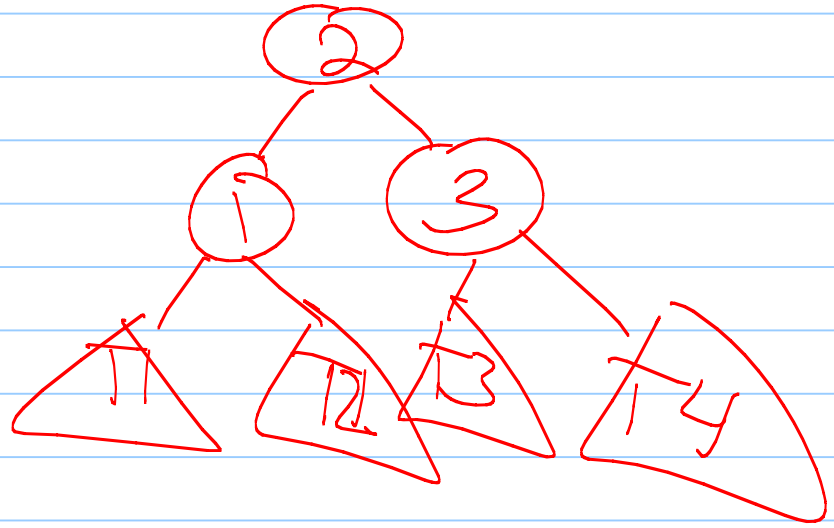
(Hint: What is inorder traversal of these in each case?)



Actual picture:

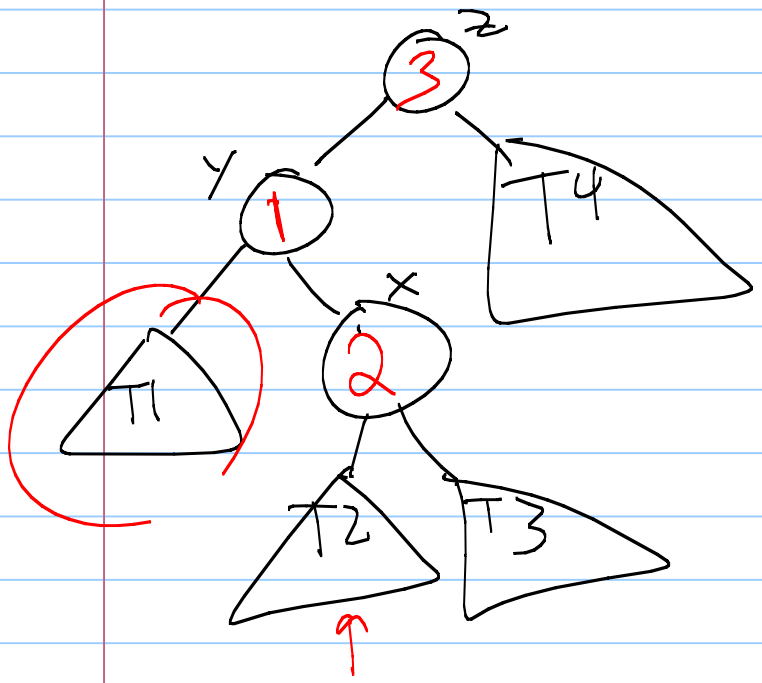


6 pointer updates

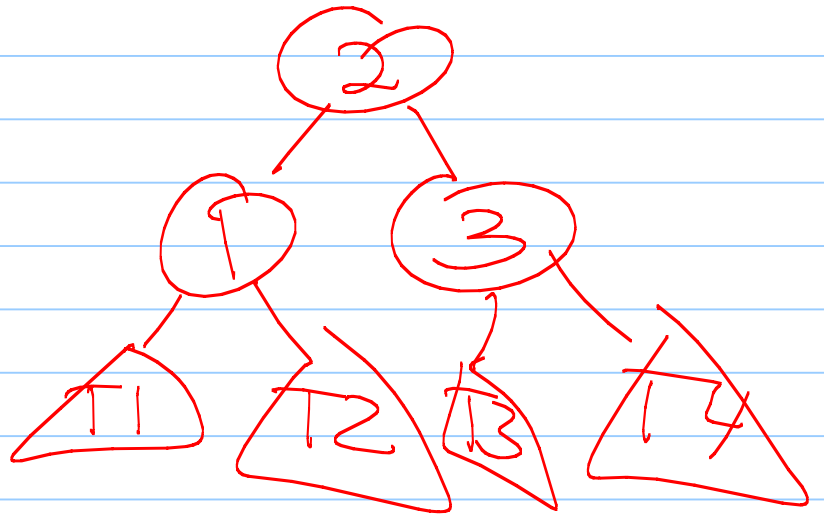


Where do the subtrees go??

Another

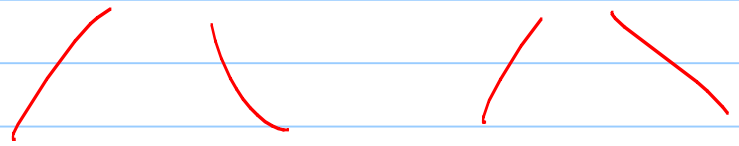


?



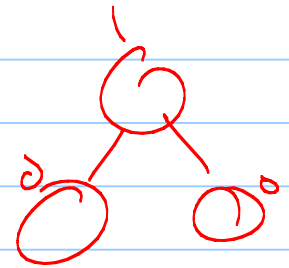
Any way you do this, "2" becomes the root of the new subtree, with "1" to the left & "3" to the right!

What about T1, T2, T3, & T4?



Insert new leaf.

Update parent's height.



while ~~*~~ height of ~~parent~~ changed.

move up & update parent height

if children's heights differ by more than one, ↵

this is ≥1.

↳ rebalance

Key operation: Pivot