CSE 326: Data Structures
AVL Trees
Node Deletion

Brian Curless

Spring 2008

Announcements

Project 2, Phase A: **due tonight**.

Project 2, Phase B: **now assigned**.

Homework #3: due at beginning of class on Friday.

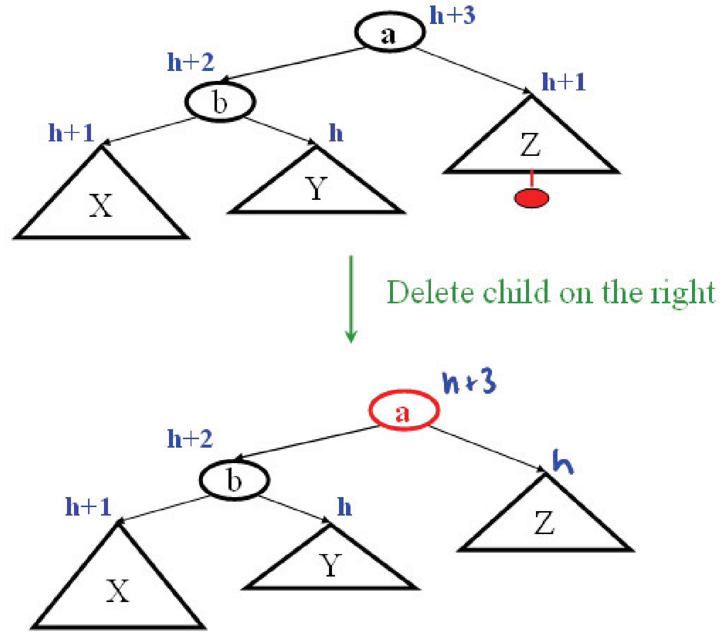
AVL Deletions

Deletions in AVL trees can be handled in a similar way as insertions, though it's a little more complicated.

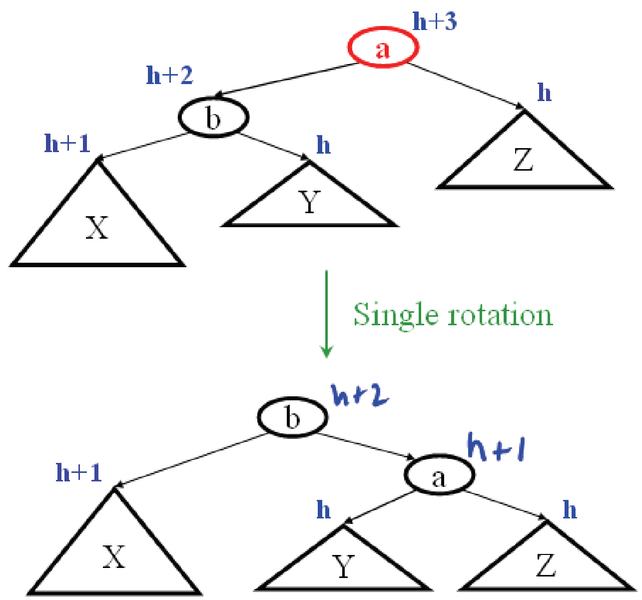
First, do a standard BST deletion. Keep track of where a leaf node was (ultimately) removed and fix up the heights above that point.

If the deletion created an imbalance, apply a suitable fix to the tree…

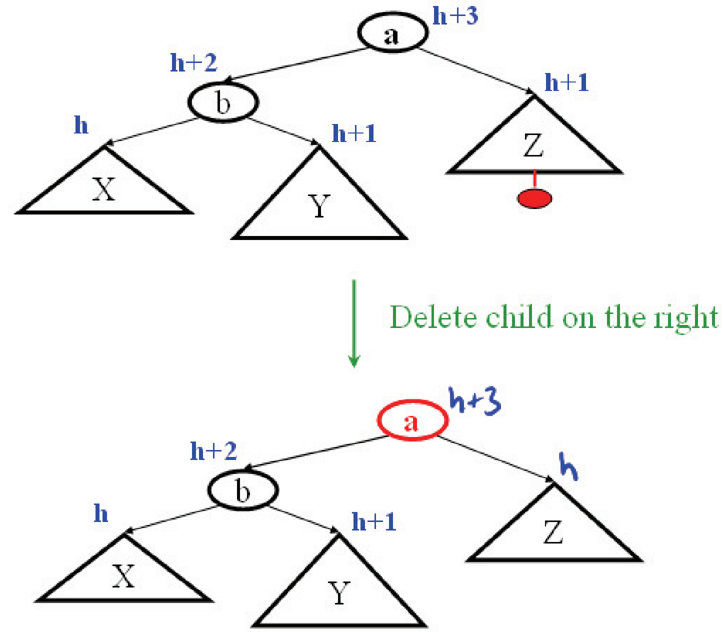
Case #1: deletion → left-left



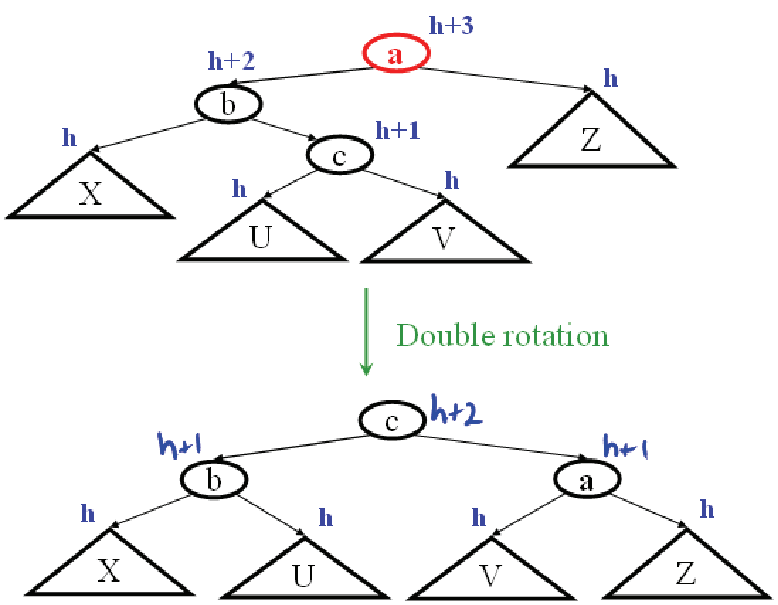
Case #1: deletion → left-left



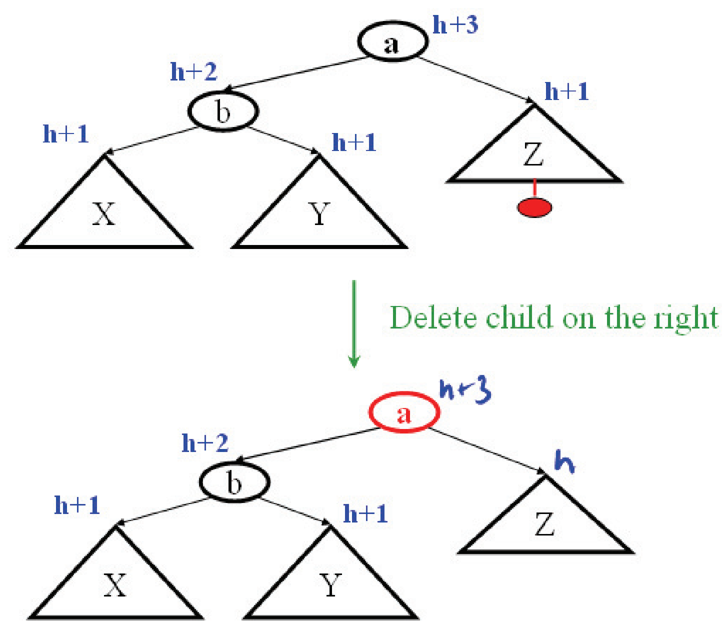
Case #2: deletion → left-right



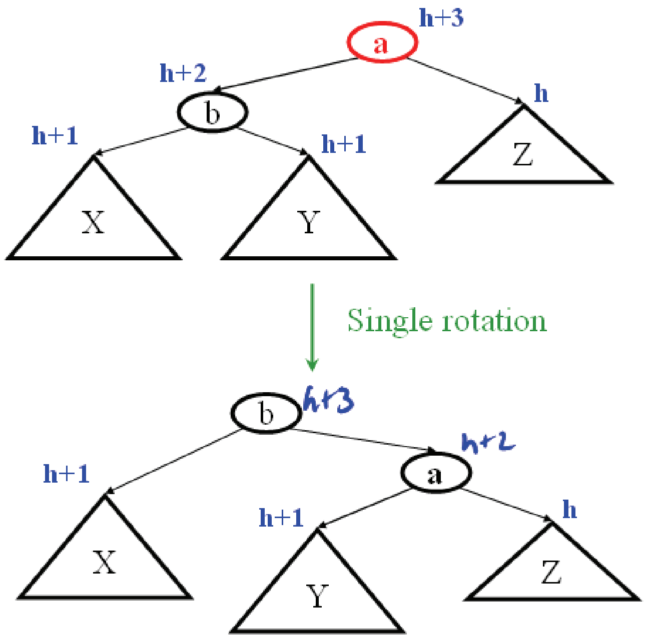
Case #2: deletion → left-right



Another deletion case?



Another deletion case?



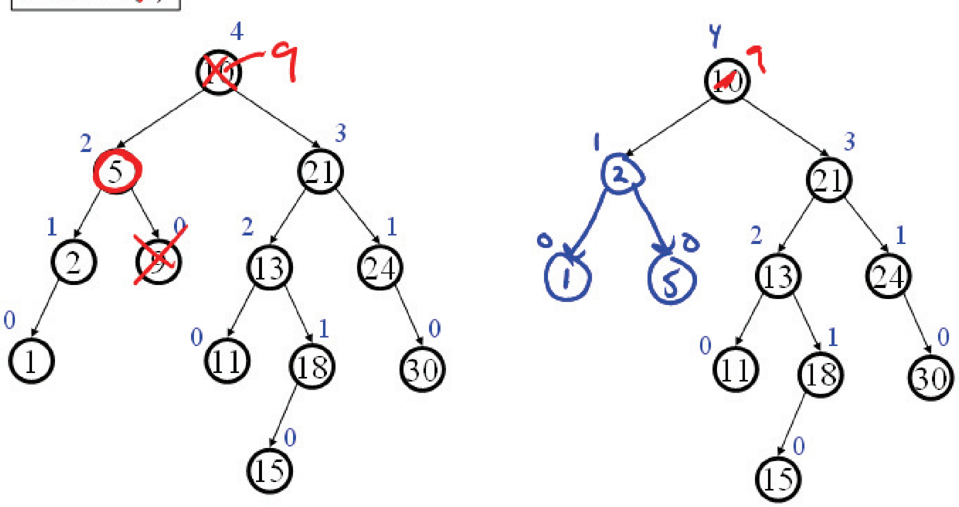
Other cases

There are of course two more mirror image cases: right-left and right-right.

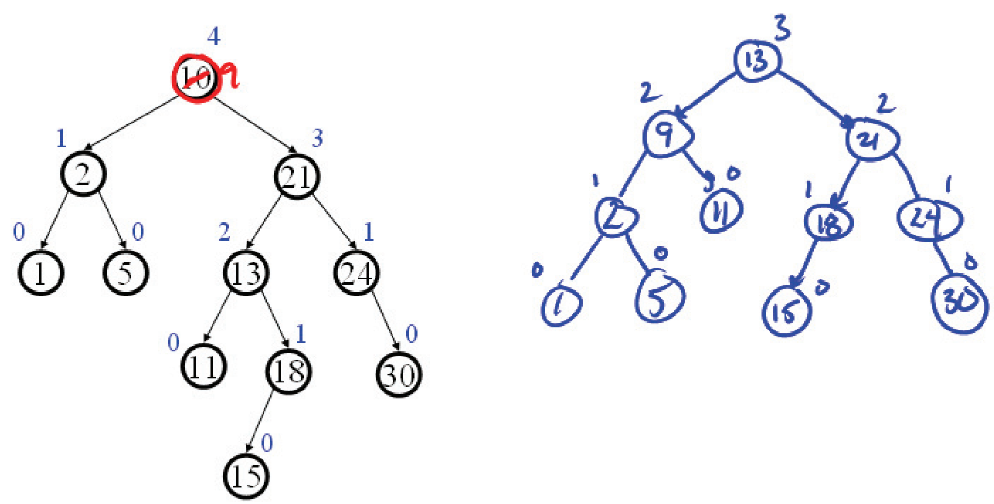
Let's try an example…

Deletion example

remove(9) 10



Deletion example



Finishing up deletions

After a deletion, we update the heights and fix the first problem we find. Then…?

Keep going up, looking for problems

What is the complexity of doing a deletion?

$O(\log n)$