

CSE 326: Data Structures

AVL Trees

Node Deletion

Brian Curless
Spring 2008

1

Announcements

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

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

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

2

AVL Deletions

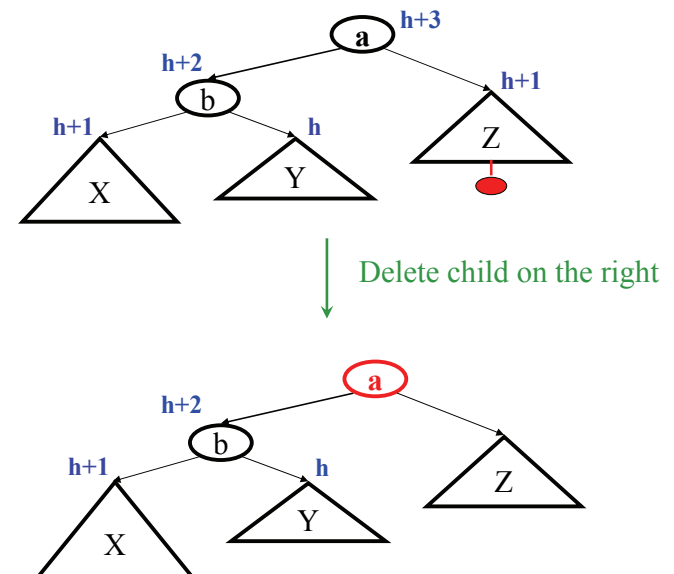
Deletions in AVL trees can be handled in a similar way as insertions, though it's a little more complicated.

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...

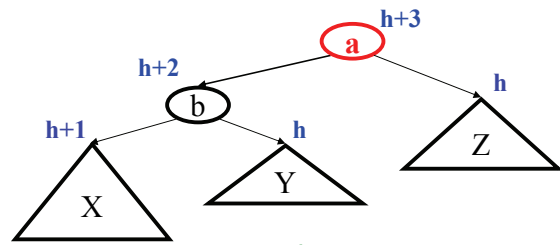
3

Case #1: deletion → left-left

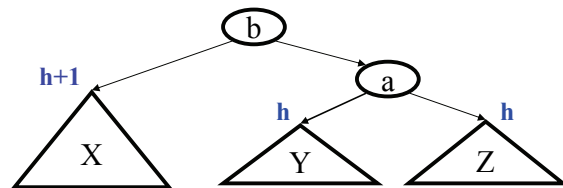


4

Case #1: deletion → left-left

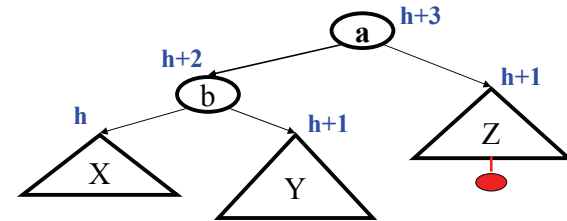


Single rotation

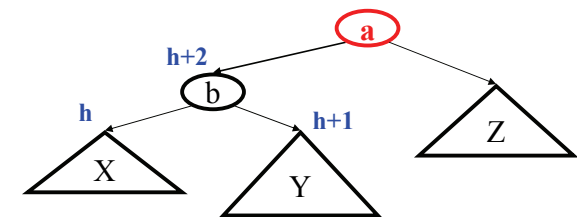


5

Case #2: deletion → left-right

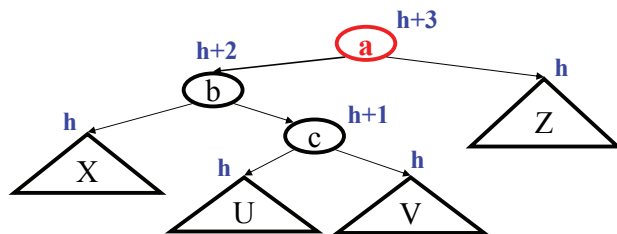


Delete child on the right

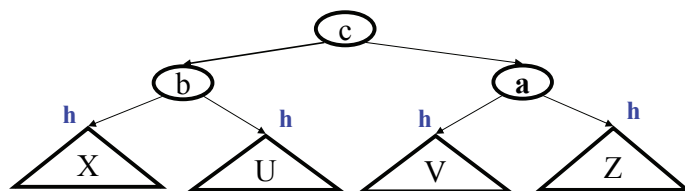


6

Case #2: deletion → left-right

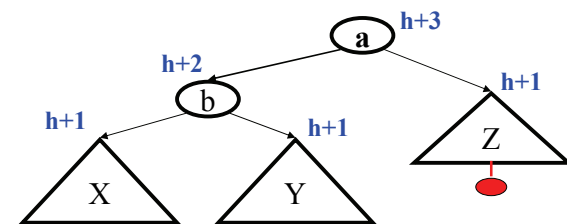


Double rotation

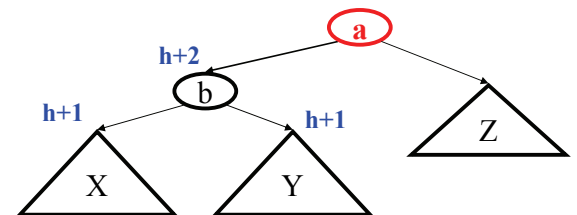


7

Another deletion case?

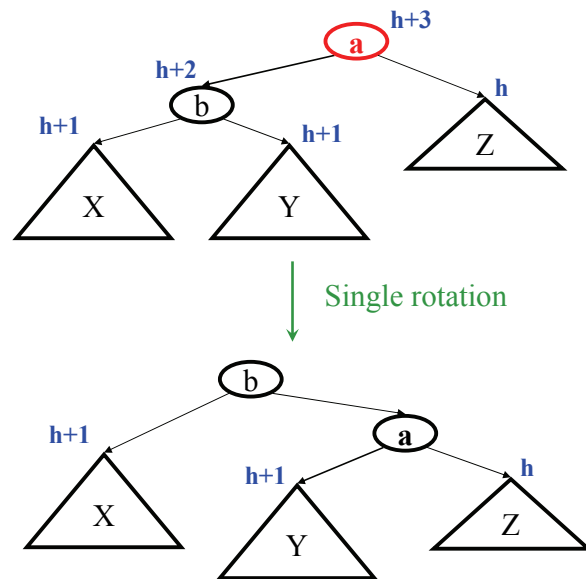


Delete child on the right



8

Another deletion case?



9

Other cases

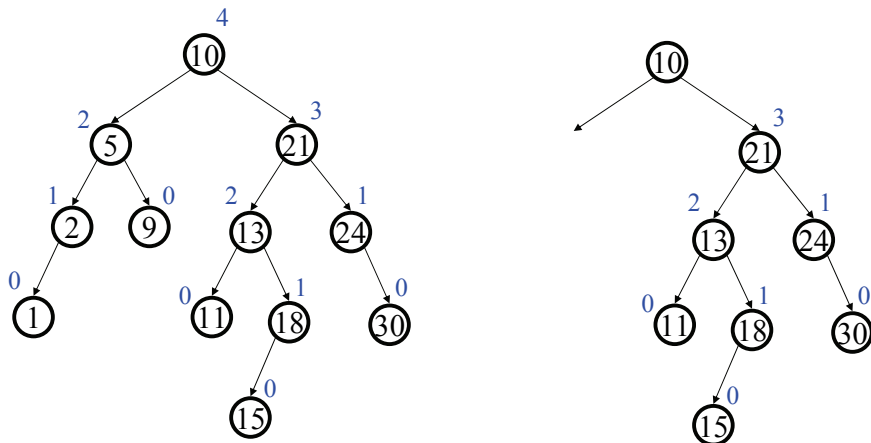
There are of course two more mirror image cases:
right-left and right-right.

Let's try an example...

10

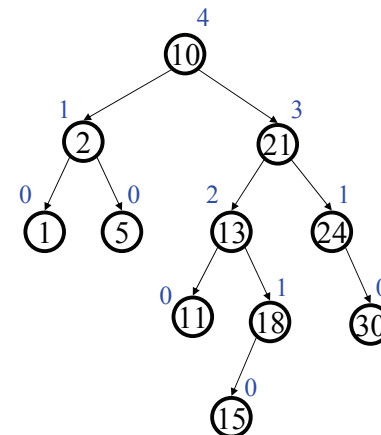
Deletion example

remove(9)



11

Deletion example



12

Finishing up deletions

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

What is the complexity of doing a deletion?