

CPSC 320: Intermediate Algorithm Design & Analysis

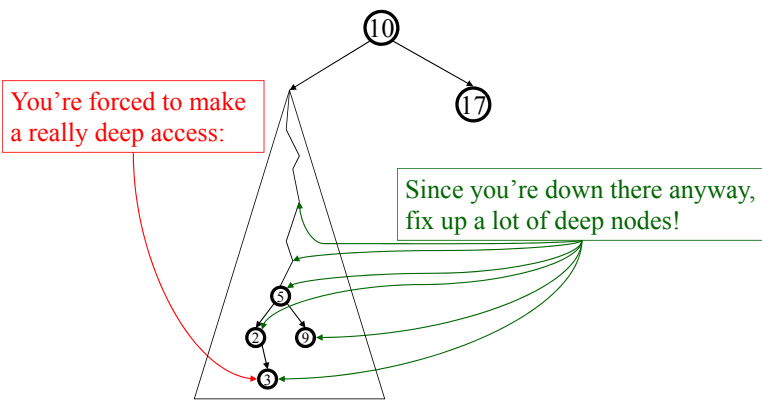
Splay Trees (for Amortized Analysis)
Steve Wolfman

1

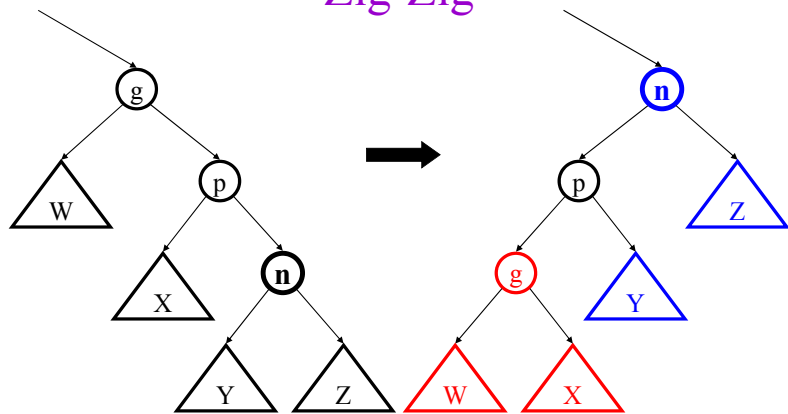
Splay Trees

- Problems with AVL Trees
 - extra storage/complexity for height fields
 - ugly delete code
- Solution: splay trees
 - blind adjusting version of AVL trees
 - amortized time for all operations is $O(\log n)$
 - worst case time is $O(n)$
 - insert/find always rotates node *to the root!*

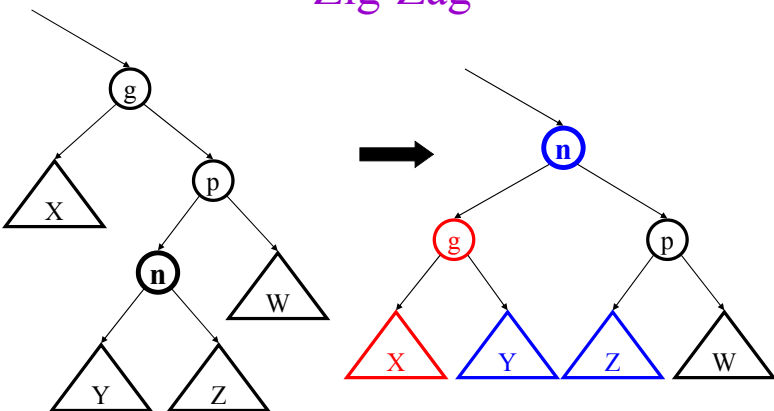
Idea



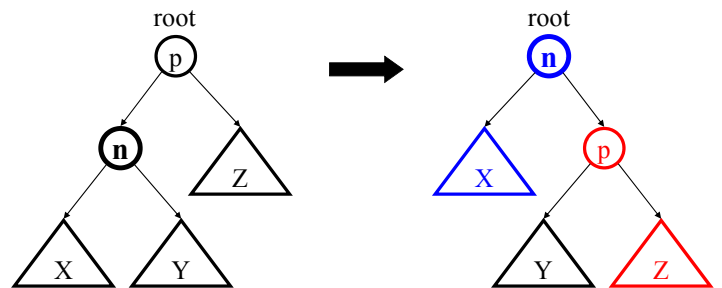
Zig-Zig



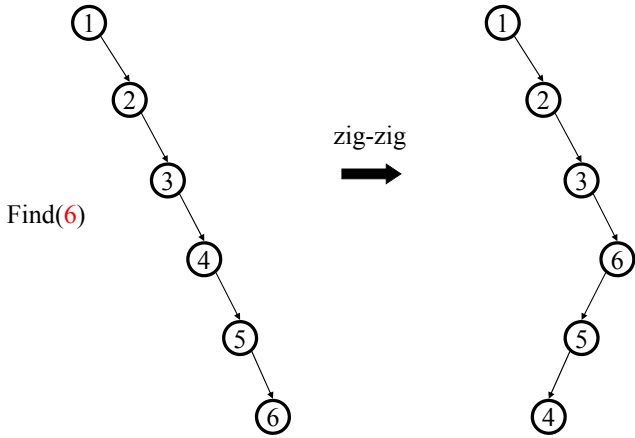
Zig-Zag



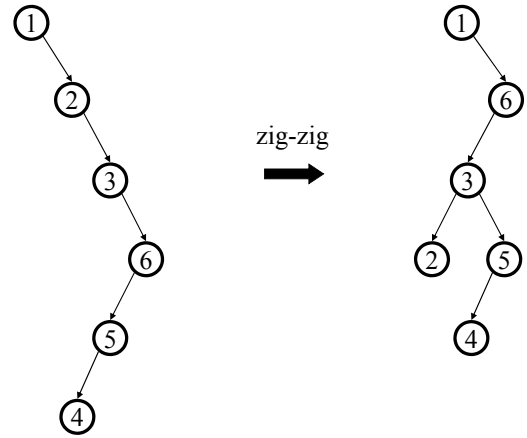
Zig



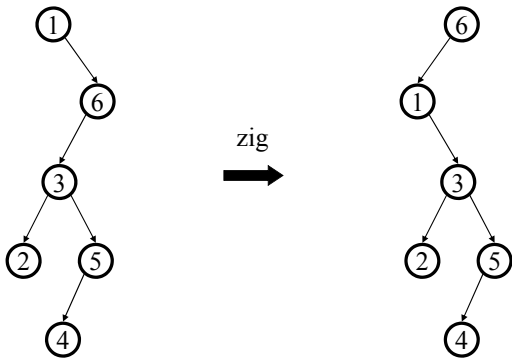
Splaying Example



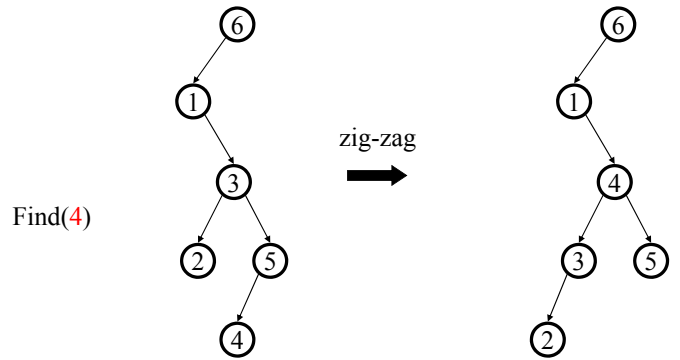
Still Splaying 6



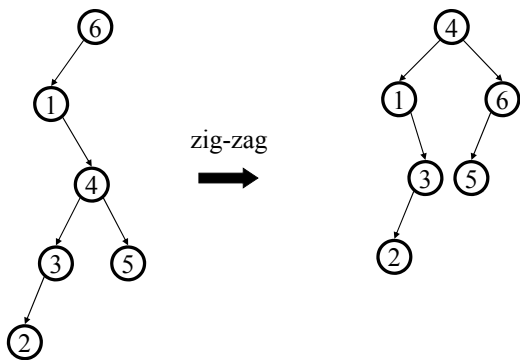
Almost There, Stay on Target



Splay Again



Example Splayed Out



How do you actually do Insert/Delete?

You do a splay or two plus some slightly tricky stuff that's still far easier than AVL, 2-3, B, B+, or Red-Black trees. Not especially relevant here.