Name: _____

Today's Date: _____

Today's Goals

- Explain how Splay and AVL trees maintain balance
- Describe the tradeoffs between Splay and AVL trees
- Explain how AVL trees have amortized performance.

Today's Question(s)

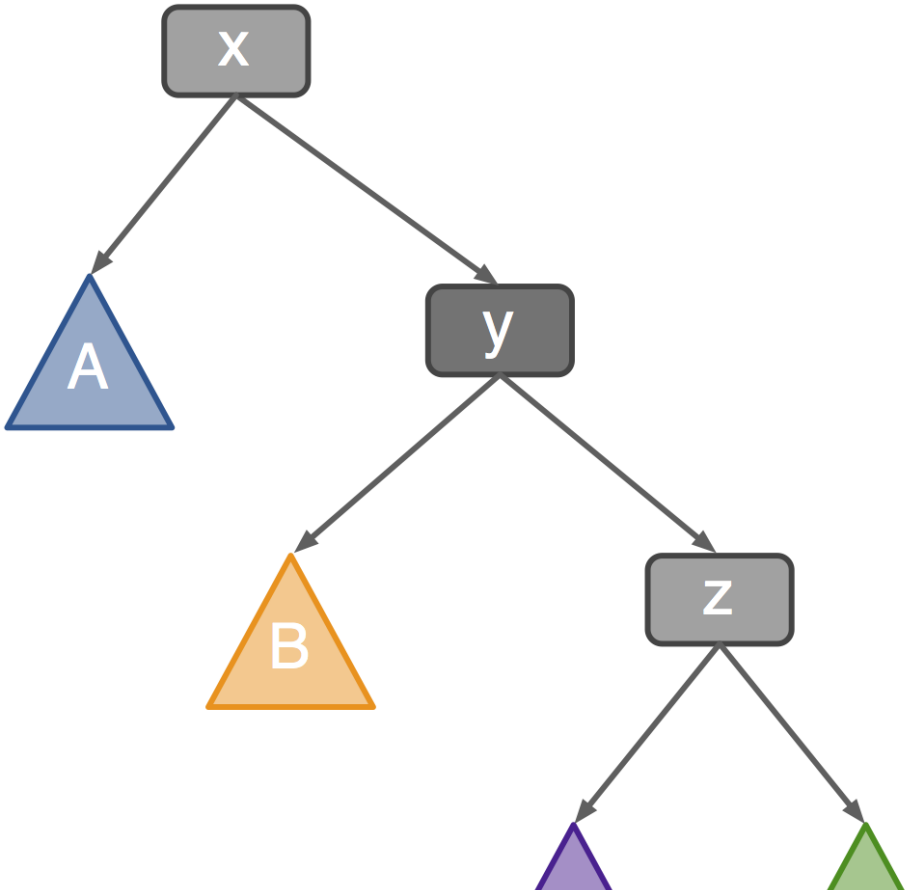
Of the trees we've seen before, which give:

- *guaranteed* constant time insert and lookup

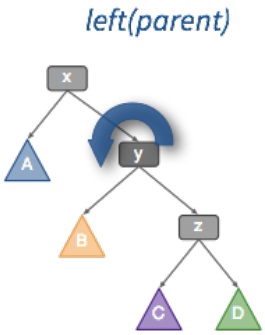
- *expected* constant time insert and lookup

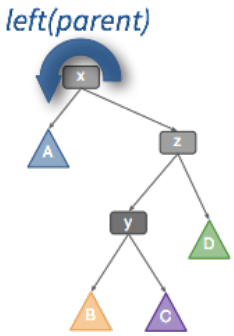
Lingering Questions

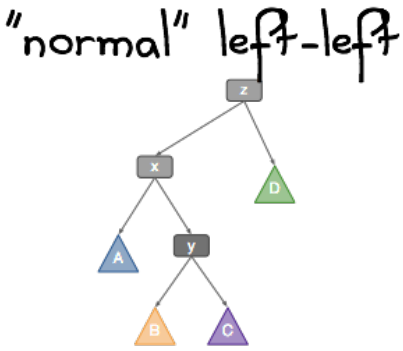
How can we bring z to the root?

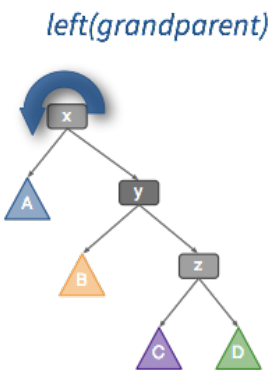


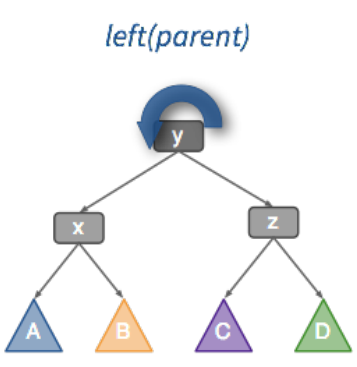
Splay rotations

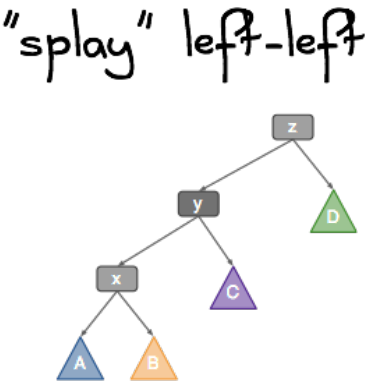












Splay Trees: Insert and Lookup

Always insert at the root, but use different rotations

- Never perform a regular left-left rotation — use ...
- Never perform a regular right-right rotation — use ...

For lookup, bring found item to root using ...

Applet demo

(Applet is great for gaining intuition, but off-limits for the final.)

Splay Trees: Properties

Splay trees are a **self-adjusting** data structure

- Recently added or accessed items migrate to the top of the tree
- Lookups balance the tree, where necessary

Advantages

-
-
-

Disadvantages

-

Well now my son is my grandson and my
Daughter is my ma and my
Granddaughter's my sister and my
Dad is my son and now my
Brother is my grandson and my
Mom is my daughter and my
Grandson is my brother and my
Son is my new pa

I'm a node in a tree
This is not about a person
It's a splay rotation on a BST
– A musical interlude from Dani Demas (CS 70, Spring 2014)

Listen Here!

AVL Trees

(named after their inventors, G.M. Adelson-Velsky and E.M. Landis)

Balance conditions

- Each subtree is balanced
- Height of each node's subtrees differs by at most 1

Advantages

- Fast find — keeps tree height minimized
- No risk of ending up with a "bad" tree

Disadvantages

- Rigid balance requirements slow down insert and delete
- Need to keep track of subtree heights