CS 367 - Introduction to Data Structures
Thursday, March 31, 2016

Homework 7 due 10 pm tomorrow, April 1st

Program 4 due 10 pm Sunday, April 17th

Last Time

Binary Search Tree (BST)

- BSTnodes
- BST class
- implementing print
- implementing lookup, insert, delete
- complexities of BST methods

CS Options/Courses

Today

Classifying Binary Trees

Balanced Search Trees

Red-Black Trees

- tree properties
- print, lookup
- insert

Next Time

Read: start *Graphs*

Finish Red-Black Trees

ADTs/Data Structures Revisited

Graphs

- terminology

Classifying Binary Trees

Full

Complete

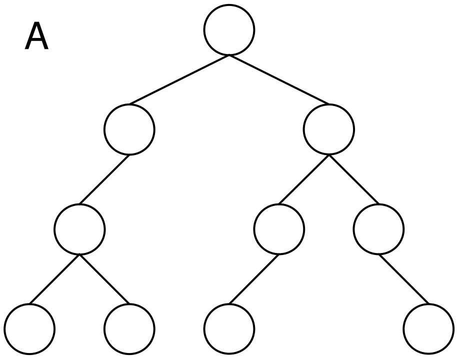
Height-balanced

Balanced

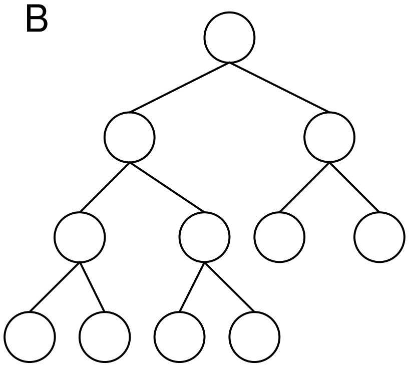
Practice - Classifying Binary Trees

→ **Identify which trees below are full, complete and/or height balanced.**

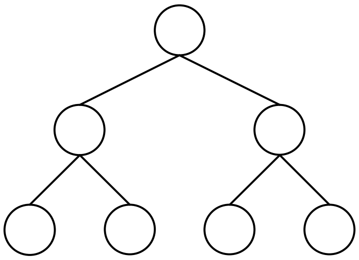
A



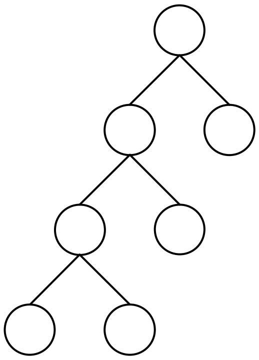
B



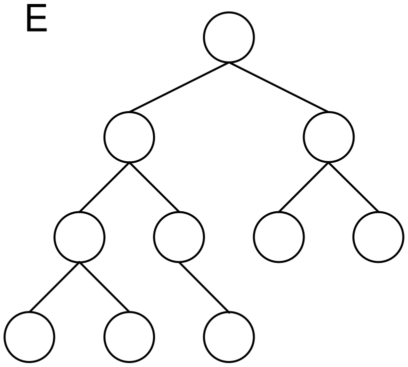
C



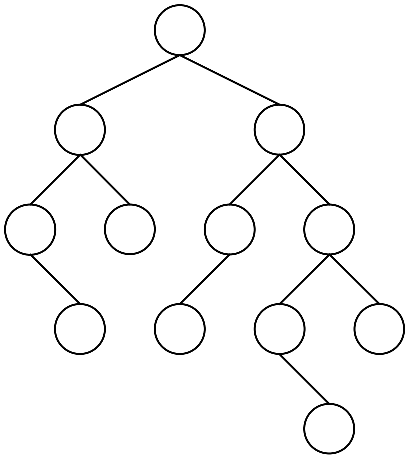
D



E



F



Balanced Search Trees

Goal:

Idea:

AVL

BTrees

Red-Black Trees (RBT)

RBT:

Example:

Red-Black Tree Properties

root property

red property

black property

Red-Black Tree Operations

print
lookup

insert

delete

Inserting into a Red-Black Tree

Goal: insert key value K into red-black tree T
and ____.

If T is Empty

If T is Non-Empty

- step down tree as done for BST
- add a leaf node containing K as done for BST, and ____________________
-

→ **Which of the properties might be violated as a result of inserting a red leaf node?**

root property

black property

red property

Non-Empty Case 1: K's parent P is black

Non-Empty Case 2

Non-Empty Case 2: K's parent P is red

Fixing an RBT

Tri-Node Restructuring is done if P's sibling S is null

Recoloring is done if P's sibling S is red

Practice

→ 1. Starting with an empty RBT, show the RBT that results from inserting 7 and 14.

→ 2. Redraw the tree from above and then show the result from inserting 18.

→ 3. Redraw the tree from above and then show the result from inserting 23.

→ 4. Redraw the tree from above and then show the result from inserting 1 and 11.

→ 5. Redraw the tree from above and then show the result from inserting 20.

More Practice!

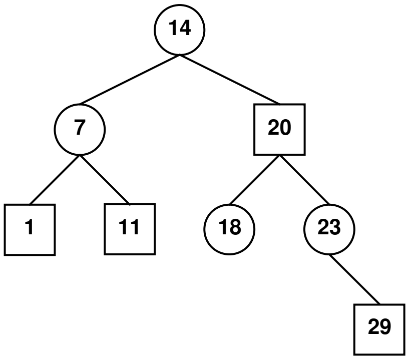
→ 6. Redraw the tree from the previous page and then show the result from inserting 29.

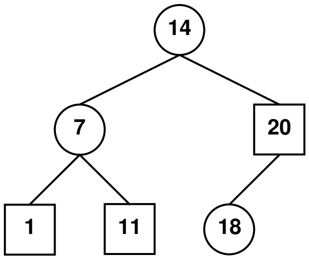
→ 7. Insert the same list of values into an empty BST: 7, 14, 18, 23, 1, 11, 20, 29

→ What does this demonstrate about the differences between a BST and RBT?

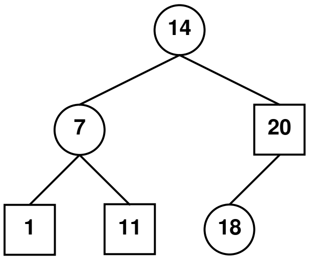
More Practice?

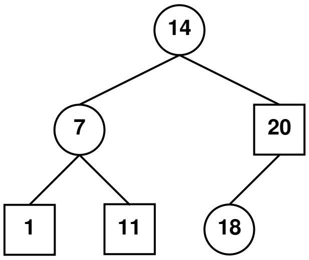
→ 8. Show the result from inserting 25 in the RBT below.





→ 9. Redraw the tree from above and then show the result from inserting 27.

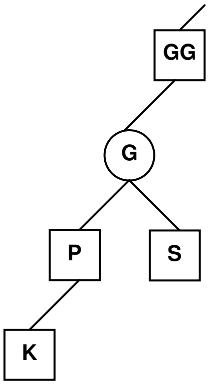




Cascading Fixes

Fixing an RBT UPDATED!

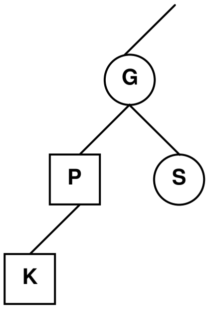
Recoloring is done if P's sibling S is red

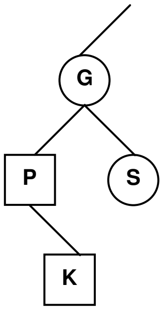


1. change P & S to black
2. if G is the root – done

   otherwise change G to red

Tri-Node Restructuring is done if P's sibling S null _______________





RBT Complexity

print

lookup

insert