CS 367 - Introduction to Data Structures
Tuesday, July 21, 2015, Lecture 21

Course website: http://pages.cs.wisc.edu/~cstapleton/367/
Piazza: https://piazza.com/wisc/summer2015/cs367/

Instructor
Cea Stapleton
cstapleton@cs.wisc.edu ; Use Piazza for course related questions.

TA
Haseeb Tariq
haseeb@cs.wisc.edu ; Use Piazza for course related questions.

Last Time
- Last week in review
- Associative Arrays/Maps
- Binary Search Trees
- delete method (finish)
- Complexities
- Balanced Search Trees (intro)

Today
- Balanced Search Trees
- Red-Black Trees
- HW6 Assigned

Next Time
- RBT example
- Hashing
Balanced Search Trees

Goal

How?

Rotations

Types
Red-Black Trees (RBTs)

Idea

Properties:

- Every node is “colored” red or black
- Root property
- Red property
- Black property
- All leaves are null and black

Example:
Operations: print, lookup, **insert, delete**

---

**Black-height of a node**

Definition:

---

Immediate consequences of RBT properties:
Inserting into a red-black tree

Insert a value V into a red-black tree T WHILE ________________________________

Trivial case: T is empty.

General case:

- Travel down and insert V as a leaf (like in a regular BST)
- Color the new node red
- What else?

Which property might be violated?
RBT Insertion (cont'd): V's parent P is red

Case 1: P's sibling S is black. In this case, **rotate**:

Case 2: P's sibling S is red. In this case, **recolor**:
Cascading Fixes

RBT insertion example

Starting with an empty RBT, insert 7, 14, 18, 23, 1, 11, 20, 29, 25, 27 (same values as before)
Complexity of RBT operations

print:

lookup:

insert:

delete: