CS367 Announcements
Tues, July 23rd, 2013

- P3 due Wed July, 31st 11:59pm

Last Time

- Binary Trees
- Binary Search Trees

Today

- BST (cont.)
- Red-Black Trees (RBTs)

Complexities of BST Methods

Problem size: N =

print:

lookup:

insert:

delete:

BST insert/delete example

Starting with an empty BST, show the BST that results from inserting:

`7, 14, 18, 23, 1, 11, 20, 29, 25, 27`

Now delete: `18,23` (using in-order successor)

Balanced Search Trees

Red-Black Trees (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

if T is empty

if T is non-empty

- **step down tree as done for BST**
- **add key K as a leaf node as done for BST, color it red**
- **restore red-black tree properties**

Case 1: K's parent P is black

Case 2: K's parent P is red

Case 2: K's Parent P is Red (cont.)

Case 2a: P's sibling S is black or null

Case 2b: P's sibling S is red

RBT Example

Starting with an empty RBT, show the RBT that results from inserting:

```
7, 14, 18, 23, 1, 11, 20, 29, 25, 27
```