CS367 Announcements
Mon, July 22nd, 2013

- H5 due Mon 6pm
- P3 due Wed July, 31st 11:59pm
- mid-semester feedback

Last Time

- Binary Trees
- Binary Search Trees

Today

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

BSTnodes

```
class BSTnode<K> {

    private K key;
    private BSTnode<K> left, right;


    public BSTnode(K key, BSTnode<K> left, BSTnode<K> right) {
        this.key = key;
        this.left = left;
        this.right = right;
    }


    public K getKey() { return key; }
    public BSTnode<K> getLeft() { return left; }
    public BSTnode<K> getRight() { return right; }


    public void setKey(K newK) { key = newK; }
    public void setLeft(BSTnode<K> newL) { left = newL; }
    public void setRight(BSTnode<K> newR) { right = newR; }
}
```

BST Class

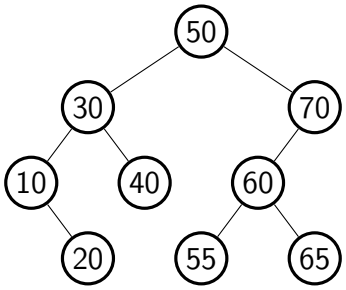
```
public class BST<K extends Comparable<K>> {

    private BSTnode<K> root;

    public BST() { root = null; }

    public void insert(K key) throws DuplicateException{



    }

    public void delete(K key) {



    }

    public void boolean lookup(K key) {



    }

    public void print(PrintStream p) {



    }
    ...
}
```

Implementing insert

```
private BSTnode<K> insert(BSTnode<K> n, K key) throws DuplicateException {
```

High-level algorithm

Inserting into a BST



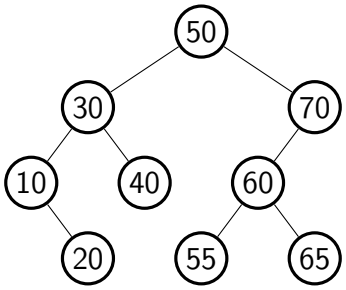
Insert: 5, 27, 90, 73, 57

Implementing delete

```
private BSTnode<K> delete(BSTnode<K> n, K key) {
```

High-level algorithm

Deleting from a BST



Delete: 40, 20, 70, 50