CS367 Announcements
Thu, July 18th, 2013

- P2 due Today, Wed 11:59pm
- H5 due Mon 6pm

Last Time

- Trees (Cont.)

Today

- Binary Trees
- Binary Search Trees

Classifying Binary Trees

full:

complete:

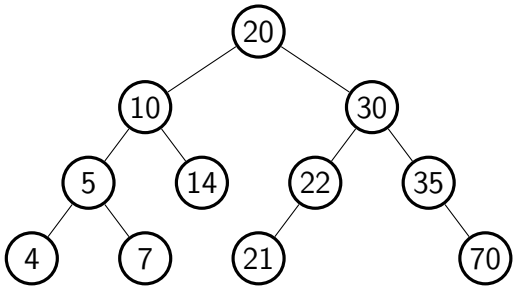
height-balanced:

balanced:

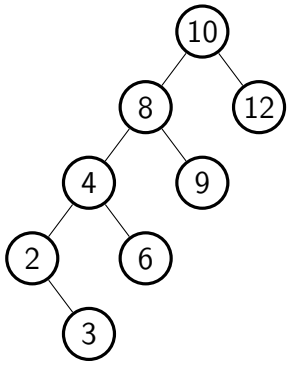
Binary Search Tree (BST)

Classifying Binary Trees

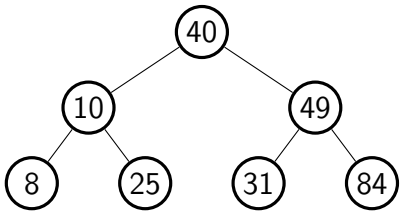
(A)



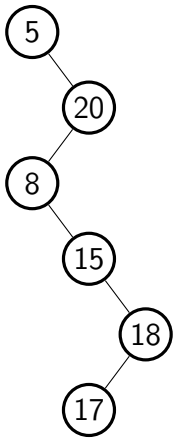
(B)



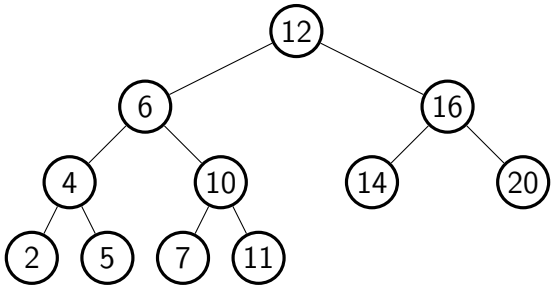
(C)



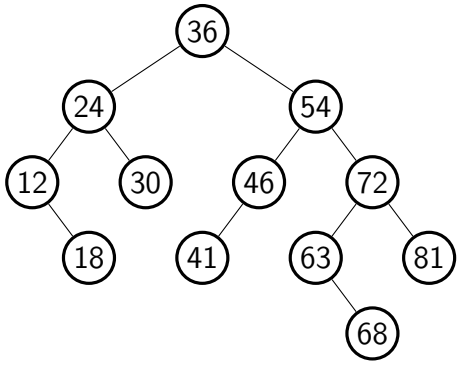
(D)



(E)



(F)



Comparable¡T¿ Interface

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 print

Psudo-code algorithm

Actual code (in BST.java)

```
public void print(PrintStream p) {
    print(root, p);
}

private void print(BSTnode<K> n, PrintStream p) {
```

Implementing lookup

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

Psudo-code algorithm