Section 9

CSCI E-22

Will Begin Shortly

Binary Search Trees

- Recall that a *binary search tree* is a special binary tree in which the keys are kept in order. In particular, we call a tree a binary search tree if it satisfies the *binary search tree property*.

- A tree satisfies the *binary search tree property* if, for a root node with key *k*...

Insert the following sequence of keys into an empty binary search tree:

`15, 23, 20, 10, 13, 6, 18, 35, 23 (a duplicate), 9, 24`

What will the tree look like after deleting the following items?:

`6, 15, 20`

Binary Tree Methods

We want to write a method which counts the number of leaf nodes in a given tree. Leaf nodes are defined as nodes which have no children.

Assume we have a node `root` which is the root node of a tree. How can we recursively define the number of leaf nodes in that tree?

```
public int numLeaves(Node root) {
```

2-3 Trees

Insert the following sequence of keys into an empty 2-3 tree:

`15, 23, 20, 10, 13, 6, 18, 35, 27, 9`

Remember:

- A 2-3 tree is a balanced tree in which:
  - *all* nodes have equal-height subtrees (perfect balance)
  - each node is either
    - a **2-node**, which contains one data item and 0 or 2 children
    - a **3-node**, which contains two data items and 0 or 3 children
  - the keys form a search tree

Insert the following sequence of keys into an empty 2-3 tree:

15, 23, 20, 10, 13, 6, 18, 35, 27, 9

B-Trees

We want to construct a B-tree where *m* = 2 for the following keys, in the specified order, from left to right:

51, 3, 10, 77, 20, 40, 34, 28, 61, 80, 68, 93, 90, 97, 14

*Remember, with B-trees:

- A B-tree of order *m* is a tree in which each node has:
  - At most 2*m* entries (and, for internal nodes, 2*m*+1 children)
  - At least *m* entries (and, for internal nodes, *m*+1 children)
  - Exception: the root node may have as few as 1 entry
  - A 2-3 tree is essentially a B-tree of order 1

We want to construct a B-tree where $m = 2$ for the following keys, in the specified order, from left to right:

51, 3, 10, 77, 20, 40, 34, 28, 61, 80, 68, 93, 90, 97, 14