CSCI 136:
Data Structures
and
Advanced Programming

Lecture 22-2

Trees, part 2

Instructors: Dan & Bill

Williams

Outline

Binary search trees

Binary search tree

A **binary search tree** is a binary tree that maintains the **binary search property** as elements are added or removed. In other words, the **key** in each node:

- must be ≥ any **key** stored in the left subtree, and
- must be ≤ any **key** stored in the right subtree.

As with other ordered structures, order is maintained **on insertion.**

Key, Value nodes

Note that I said **key** instead of **element**.

Storing a **key** and a **value** in each node allows the greatest flexibility when arranging a tree. I.e., the key type K need not be the value type V.

Restriction: keys must be **comparable** in some way (e.g., `Comparable<K>` or `Comparator<K>`).

Example

Insert the following elements: `71,20,27,17,91,14,87`

Assume K and V are the same.

Example

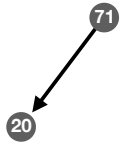
Insert the following elements: `71,20,27,17,91,14,87`

Assume K and V are the same.

Example

Insert the following elements: `71,20,27,17,91,14,87`

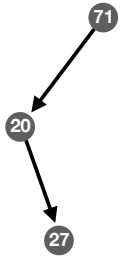
Assume K and V are the same.



Example

Insert the following elements: `71,20,27,17,91,14,87`

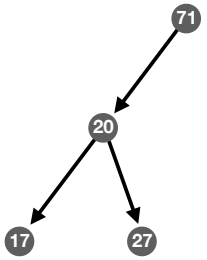
Assume K and V are the same.



Example

Insert the following elements: `71,20,27,17,91,14,87`

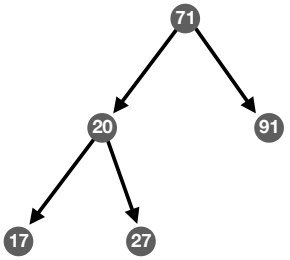
Assume K and V are the same.



Example

Insert the following elements: `71,20,27,17,91,14,87`

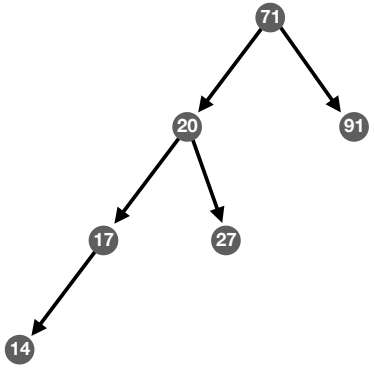
Assume K and V are the same.



Example

Insert the following elements: `71,20,27,17,91,14,87`

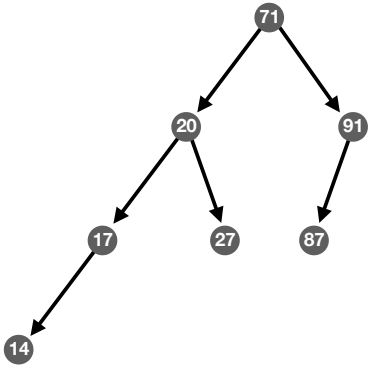
Assume K and V are the same.



Example

Insert the following elements: `71,20,27,17,91,14,87`

Assume K and V are the same.



Activity

Insert the following elements:

Assume K and V are the same.

Binary Search Tree

Let's implement this together.

Recap & Next Class

This lecture:

Binary search trees

Next lecture:

BST Big-O

Implicit BST