Lecture 19 - March 27

Binary Search Trees

BST: Search Property BST: Sorting Property BST: Constructing BST Nodes

Announcements/Reminders

- Assignment 4 (on <u>linked</u> Trees) released
- Makeup Lecture to be posted
- Lecture notes template, Office Hours, TA Contact







Building Sorted Seq. from In-Order Traversal on BST



Exercise: Checking the Search Property (1)

<u>Remember</u>: For a **BT** to be a **BST**, the **Search Property** should hold <u>recursively</u> on the <u>root</u> of each <u>subtree</u>.

In-Order: <8, 17, 21, 28, 29, 32, 44, 54, 65, 76, 80, 82, 88, 93, 97>



Exercise: Checking the Search Property (2)

<u>Remember</u>: For a **BT** to be a **BST**, the **Search Property** should hold <u>recursively</u> on the <u>root</u> of each <u>subtree</u>.

In-Order: <8, 17, 21, 28, 29, 32, 44, 54 65, 76, 80, 82, 88, 93, 97>



Visual Summary: In-Order Traversal on BST



Generic, Binary Tree Nodes

```
public class BSTNode
 private int key; key * Intel
 private E value; /* value */
 private BSTNode<E> parent; /* unique parent node */
 private BSTNode<E> left; /* left child node */
 private BSTNode<E> right; /* right child node */
                          ext. node
 public BSTNode() { ... }
public BSTNode(int key, E value) { ... }
 public boolean isExternal()
  return this.getLeft() == null && this.getRight() == null;
 public boolean isInternal() {
  return !this.isExternal();
 public int getKey() { ... }
 public void setKey(int key) { ... }
 public E getValue() { ... }
 public void setValue(E value) { ... }
 public BSTNode<E> getParent() { ... }
 public void setParent(BSTNode<E> parent) { ... }
 public BSTNode<E> getLeft() { ... }
 public void setLeft(BSTNode<E> left) { ... }
 public BSTNode<E> getRight() { ... }
 public void setRight(BSTNode<E> right) { ... }
```



Generic, Binary Tree Nodes - Traversal

```
import java.util.ArrayList;
public class BSTUtilities<E> {
 public ArrayList<BSTNode<E>> inOrderTraversal(BSTNode<E> root)
  ArrayList<BSTNode<E>> result = null;
  if(root.isInternal()) {
    result = new ArrayList<>();
    if(root.getLeft().isInternal)
      result.addAll(inOrderTraversal(root.getLeft()));
    result.add(root);
    if(root.getRight().isInternal)
      result.addAll(inOrderTraversal(root.getRight()));
  return result;
```



Tracing: Constructing and Traversing a BST



(28, "chr

Sorted TA-0010 SEG