

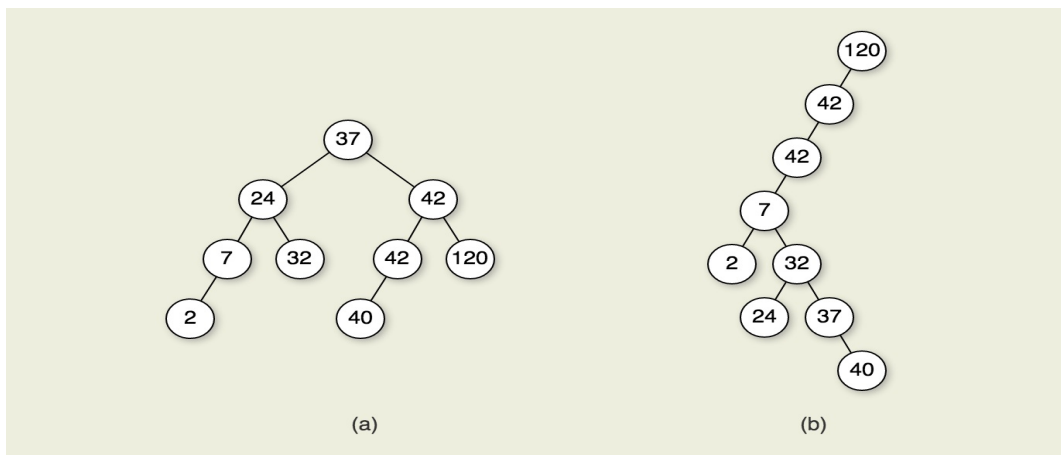
Homework 9

Submission Instructions

1. Submit completed **BST.java** through in Blackboard.
2. DO NOT forget to mention your full name in the Java class documentation.
3. submit a filled in **cover sheet** in Blackboard.

Introduction

A binary search tree (BST) is a binary tree that conforms to the following condition, known as the binary search tree property. All nodes stored in the left subtree of a node whose key value is K have key values less than or equal to K . All nodes stored in the right subtree of a node whose key value is K have key values greater than K . Figure 11.4.1 shows two BSTs for a collection of values. One consequence of the binary search tree property is that if the BST nodes are printed using an inorder traversal, then the resulting enumeration will be in sorted order from lowest to highest.



The figure shows two Binary Search Trees for a collection of values. Tree (a) results if values are inserted in the order 37, 24, 42, 7, 2, 40, 42, 32, 120. Tree (b) results if the same values are inserted in the order 120, 42, 42, 7, 2, 32, 37, 24, 40.

Download Materials

For this homework, you need to download the following starter code from Blackboard:

- BST Files:
 - BST.java - Unfinished BST class.
 - BSTNode.java
- Linked list files needed for completing the `makeOrderedList` method:
 - Link.java
 - List.java
 - LList.java

Part 1 - Implement BST methods

You need to complete the methods at the bottom of the file so that they conform to the Javadoc comments. Here are the 5 UNFINISHED methods that you need to implement:

- `public int recursiveSize()`
- `public E minElement()`
- `public E maxElement()`
- `public LList makeOrderedList()`
- `public boolean isValidBST()`

Grading Criteria

Total points: 100 points

Each of the 5 method is worth 20 points.

Don't forget

the [submission process](#) or
the [grading criteria](#) or
the [cover sheet](#).

Farzana Rahman / frahman@syr.edu