Binary Search Trees

4-8-2010
Let's look at solutions to the interclass problem.
Do you have any questions about the assignment?
Do you have any questions about the reading?
Examining the parser.
Let's begin by finishing up the code we started last time.

We need to test the parser and then perhaps put it into our drawing program/command processor.
• Sometimes we want to limit how many children a node has. One of the most commonly used trees in programming is the binary tree where no node has more than 2 children.

• The children are often called left and right.
For a binary tree there is an extra type of traversal called an in-order traversal where the node is visited between the recursion down left and right.

Equations are great examples of trees. We typically write them out in the in-order. We could just as well write them out in post-order or pre-order.
Sorted Binary Trees

• One of the best uses of binary trees is the sorted binary tree. They make a more efficient implementation of the map ADT.
• In this type of tree, we store a key and data in every node and below any node we put lesser key values to the left and greater key values to the right.
• We find elements by going down the tree always going left or right. This gives us behavior like a binary search, but the tree is more flexible because adds and removes are quite efficient as well.
The code for both adding and removing from a binary tree begins like a search that keeps track of previous (much like a singly linked list).

- The add always goes to a leaf and adds the new element to the proper side.
- The remove replaces the node we are removing with either the greatest node on the left or the smallest node on the right.
I want us to code a BST based map together.
What can go wrong with the type of binary tree that we wrote today to make it perform poorly?

Interclass Problem – Edit a Drawable so that it uses a formula for something instead of a regular double.