Binary Search Trees

4-5-2002

Opening Discussion

- What did we talk about last class?
- Snow White or a Dwarf? Due to the disk space on Snow White I made a copy of everything on it on Dwarf6 last night. Everyone should switch to working on Dwarf6. If you have newer files than what I copied let me know and we can copy the new ones.
- What is a binary search tree?

Binary Search Trees

- As I mentioned before, we often use trees that are sorted in certain ways. The most common form of a sorted tree is a sorted binary tree.
- In a sorted binary tree, every node stores a data element and everything to the left of a node is "less" than the value at that node.

Searching

- As with all data structures, one of the fundamental operations we want to be able to perform is finding data in it.
- The nature of these trees allows us to do this efficiently. We look at a node. If it is equal we return its data. If it is greater we go to the left otherwise we go to the right.
- This can be done in a loop or recursively.

Adding Data

- Adding data into a tree is very similar to a search. We follow a path to where it should be and once we are at a leaf we add it on the correct side of that leaf.
- If it is a duplicate we can handle it in a number of different ways. Allowing duplicate values can cause problems. If the data is different we can put a linked list of data in the node. Otherwise count how many have been added.

Deletion

- Deleting a node is slightly more difficult, but only slightly. First we do a search to find the node. We can't just remove it or "link around" it because it can have two children.
- If it has one child we "link around" it.
- Instead, we want to replace it with a node from below it that will work in its place. It helps if that node is "easy" to remove.

Deletion Continued

- The easy nodes to remove from any sub-tree are the smallest and largest nodes.
- The smallest node in the right sub-tree and the largest node in the left sub-tree both work as replacements for that node.
- This type of deletion does not significantly alter the total height of the tree which is a good thing.

Code

- Now let's write a sorted binary tree code and write the methods that we just discussed. This is basically what you will be doing for the first objective for assignment #5.

Minute Essay

- Compare and contrast a binary tree to a linked list. In what ways is it superior? How might it be inferior? What about compared to an array?
- The design for assignment #5 is due on Monday.