Right Threaded Trees and Statically Linked Trees

10-2-2003

Opening Discussion

- What did we talk about last class?
- Do you have any questions about the assignment?
- You really want to have a submission for assignment #2 that compiles. If your first one didn't you should resubmit as soon as you have one that does.

Code for Removing a Node

- Let's go ahead now and look at code for removing a node from a binary search tree.

Traversals without Recursion

- In some situations it can be nice to be able to traverse a tree without using recursion. Sometimes we even want to do it without implementing your own stack/queue to keep track of things.
- An example of where we would want to do this is if we have an iterator for our tree. The iterator is used in a loop so we don't a recursion stack.

Right Threaded Trees

- One way of implementing this is to do what is referred to as right threading. In a right threaded tree, right branches that would normally be null are set to point back up the tree so a right edge either points to the right child, or to that node's successor up the tree.
- The problem is telling which it is. For that we need to keep extra data in our nodes. A level value works fine here as would marking whether we have passed it before. The marking can cause other problems.

Iterators

- Perhaps the simplest way for us to do an iterator is not to right link, but instead to write successor and predecessor methods so the iterator can quickly get the next value if it know the node of the last value.
- This isn't as efficient as right threaded trees for walking, but is easier to write and maintain.

Statically Linked Trees

- We discussed doing "static linking" in a linked list using integer indexes instead of pointers. The same can be done for a tree.
- Last semester you also saw that a complete binary tree can be efficiently stored in an array. This isn't good for adding or removing though.

Keeping Sizes and other Values in Nodes

- We mentioned that we might want to keep extra information in a right threaded tree. There are also other situations where keeping extra information helps.
- If you want to do direct access by index in a binary search tree, you can store the size of each node in the nodes and do this in O(log n) time.

More Code

- Let's go through and write more code that shows some of the other things we have talked about.

Minute Essay

- What method would you use to put an iterator in your binary search tree? Do you see the value of iterators in binary search trees?
- Remember to send me a group evaluation. Some didn't for the last assignment. If you don't this time I will dock at least 10 points from your grade on the assignment.