

Trees and Binary Trees

3-27-2002

Opening Discussion

- What did we talk about last class?
- Do you have any questions about assignment #4. This is the last class you can ask them before it is due. Of course, you can send me e-mails as well.
- Alternate method of tracing mergesort.
- Worry about taking addresses (&).
- What is a tree?

What is a Tree?

- You are all familiar with what normal trees look like. In CS we use the term somewhat differently, and more formally.
- To describe trees we need some basic terminology
 - Node - an element of a tree. One node is designated as the "root"
 - Edge - a directed connection from one node to another.

Tree Criteria

- Every node, C , has exactly one incoming edge from another node, P . P is said to be the parent of child node C .
- There is a unique path from the root to any node. The number of edges on that path is called the path length. It is also called the depth of the node.
- A node with no children is called a leaf. The path length from a node to the deepest leaf is the height of that node.

More Terms

- Following the parent-child analogy, children of the same node are called siblings. We also call any node on a path below a given node a descendant and any above an ancestor.
- You might also hear the size of a node referred to as the number of descendants of a node, including itself.
- We can also define a tree as either empty, or a root with zero or more subtrees where the root connects to the roots of those subtrees.

General Tree Implementation

- In a general tree each node can have zero or more children. That is a lot of flexibility. We want a class to represent nodes. To get this flexibility we can use a linked list. Each node has pointers to a first child and the next sibling.
- In C++ it might be just as easy to have the child member be a Vector of Node*. File systems are a good example of this.

Traversals

- As with any data structure one of the things you want to be able to do is to traverse through all the elements.
- Think for a while about how you would do this? There is even a question about the order you traverse them in. Do you want to process a node before you process its children or after? If before we call it a preorder traversal. Otherwise it is a postorder traversal.

Traversals and Recursion

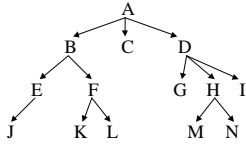
- The simplest way to do a traversal is through recursion. If you want to do it with a loop you have to implement a data structure to store some nodes or have the tree specially set up.
- The traverse function takes a node and calls itself once with each child node. It also does whatever the visit operation is.

Binary Trees

- 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. Your book has a fair bit more discussion of binary trees that we won't go into right now but you should look at.

Minute Essay

- Write the order the nodes would be visited with the following tree in a prefix traversal and then for a postfix traversal.



- There is no class on Friday.
