CSCI 262 Lecture 19 – Binary Trees

Outline

- A new linked data structure – binary tree
    - Two node pointers, not one
    - Hierarchical "tree" structure
        - Root node with two "child" nodes, each representing binary trees (may be null, representing an empty tree)
        - The connections between nodes in a tree are called "edges"
- Min/max height of binary tree
- Implementation
    - We are introducing this as an underlying data structure enabling other data structures
    - Work with bare nodes – cannot really talk about specific encapsulated implementations until we consider particular applications for the binary tree (e.g., binary search tree, decision tree, etc. all have different behaviors)
- Traversals – recursive algorithm to visit all nodes
    - pre-/in-/post-order
    - Many applications

More Info

There are different ways to graphically represent a tree, with different advantages, but the various approaches can lead to confusion. One approach that often occurs in algorithms textbooks is to require all internal nodes to have two children, and all leaf nodes contain no data. This provides some advantages for analysis – without actually changing the "big O" properties – and also for writing some algorithms, but it isn't how you would usually implement a tree in practice, as it doubles the number of nodes overall.

Self Check

1. Tree nodes that have children get a special name; they are called ______________________________.
2. Tree nodes with no children are called ______________________________ or ______________________________.
3. The number of edges between a node and the root node of a tree is called the ______________________________ of the node.
4. The maximum depth of any node in a tree gives us the ______________________________ of the tree.

For Further Practice

Following the examples of traversal applications in the slides, implement the function

```
T min_value(binary_tree_node<T>* tree)
```

which (assuming the node stores comparable elements) returns the minimum value stored in the tree.