Algorithms and Data Structures I - Fall 2018 - Programming Assignment 5

Lab 5: General Tree Traversal

Directions: This assignment will be demoed in lab on Thursday, November 29th or Friday, November 30th.

Assignment Details:

Your task in this lab is to implement a `GeneralTree` class in C++. Note that your implementation *must support nodes with more than two children*, as opposed to the `BinarySearchTree` code from the prior lab, which supported a maximum of two child nodes. Your `GeneralTree` class should define the following public methods:

1. `void addEdge(int n1, int n2)` - Insert an edge in the tree from node with id `n1` to node `n2`
2. `void preorder()` - Traverse the tree using preorder traversal. Print (to standard output) the ID of every node encountered
3. `int count()` - return the total number of nodes in the tree
4. `int height()` - return the height of the tree

Input will be provided in the form of an edge list, specified in terms of node IDs. A node ID is an integer values which ranges from 1 to n. An edge from node $n1$ to $n2$ implies that $n2$ is a child node of $n1$.

For a detailed discussion of general trees and tree traversal, see chapter 6 of A Practical Introduction to Data Structures and Algorithm Analysis. In particular, see sections 6.1.2 (General Tree Traversals)

Deliverable:

C++ code that compiles to a single executable. This executable should accept as input an edge list (to be provided). Print to standard output the string "Preorder:" followed by ID of every node based on preorder traversal. On separate lines, print the total number of nodes prefixed by "Count:" and the height of the tree, prefixed with the string "Height:"

For example, suppose that the given edge list is:

```
(7 5) (1 3) (5 8) (5 1) (1 6) (1 4) (7 9) (7 2)
```

Note that depending on how you build your tree, your preorder may differ from mine.

```
$ ./a.out <edge list>

Preorder: 7 5 8 1 3 6 4 9 2

Count: 9

Height: 3
```