COP3530

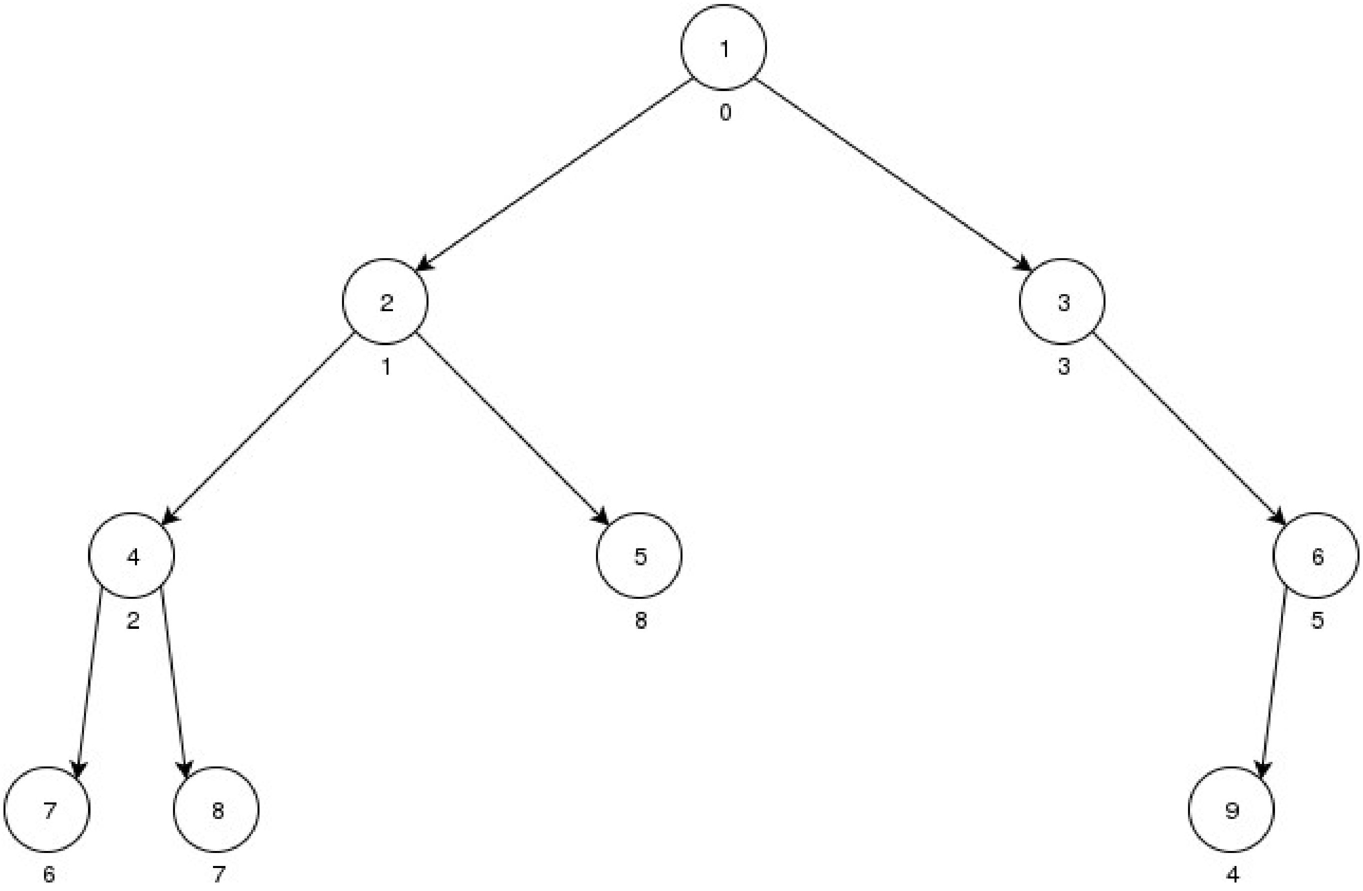
Programming Assignment #3

Problem 1: Pre/In/Post-order Traversal

For this problem you should write a program that reads a binary tree from the input and prints the preorder, inorder, and postorder traversals of the tree.
Each node of the tree contains a number between 0 and 1000000, inclusive.

INPUT:

On the first line, N, showing the number of nodes in the tree. For the next N lines, the i^{th} line contains the data and the left and right children of the i^{th} node. For example, if the 5^{th} line reads "154 4 6", it means that the 5^{th} node contains 154, its left child is the 4^{th} node (the node which has its information on the 4^{th} line), and its right child is the 6^{th} node (the node which has its information on the 6^{th} line). The number "-1" shows that the child does not exist, for example, "12 -1 -1" shows a node that contains 12 and has no children (i.e. is a leaf).
The numbers are 0 based, and the 0^{th} node is always the root of the tree. For an example, in the tree below, the numbers in the nodes are the data, and the numbers under the node show which line contains their information:



Same tree denoted in the described input format:

```
9
1 1 3
2 2 8
4 6 7
3 -1 5
9 -1 -1
6 4 -1
7 -1 -1
8 -1 -1
5 -1 -1
```

OUTPUT:

The output consists of 3 lines. The first line should contain the preorder traversal of the tree, the second line should contain the inorder traversal of the tree, and the third line should contain the postorder traversal of the tree. In order to print the traversals, print the data in the nodes as you visit them separated by a single space. The output for the given input is:

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

Problem 2: Levelorder Traversal

Similar to the previous problem, your program should read a tree, as described the first problem, and print out its levelorder traversal for the output.

INPUT:

Similar to the input for the previous problem.

OUTPUT:

Similar to the output for the previous problem, but in this problem the output will only have 1 line, which should contain the levelorder traversal of the tree.
Correct output for the input given in the previous problem:

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

Problem 3: Tree Reconstruction

For this problem, you have to reconstruct a tree using its postorder and inorder traversals, and print its levelorder for the output.

INPUT:

On the first line you can read the number of elements in the tree. On the second line you can read the postorder traversal of the tree, in a format similar to the output format of problem 1, i.e. each node's data is printed separated by a single space. On the third line you can read the inorder traversal of the tree in the same format.

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

OUTPUT:

The output should be one line containing the levelorder traversal similar to problem 2.

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