

COP3530

Assignment #4

Binary Tree Traversal

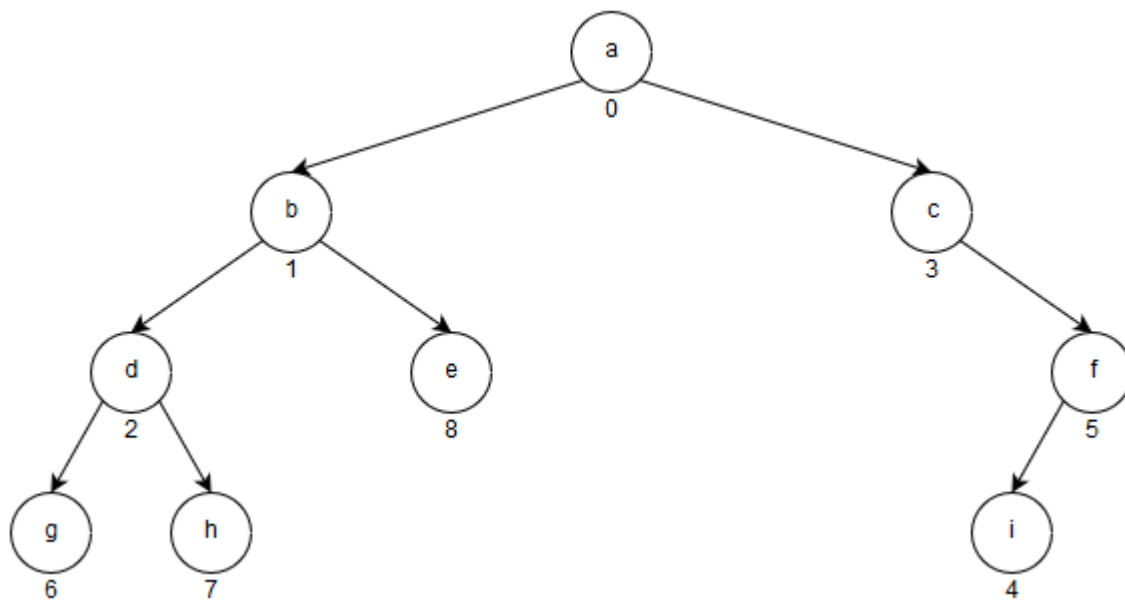
Problem 1. Pre/In/Post-order traversal

For this problem you should implement a program that reads a binary tree from the input and prints the postorder, inorder and preorder traversals of the tree.

Each node of the tree contains a string of lower case alphabetical characters.

Input format.

On the first line, there will be a single number, N, showing the number of nodes in the tree. On the next N lines, the i^{th} line contains the data and the left and right child of the i^{th} node, for example if the 5th of the N lines reads `abc 4 6` it means that the node number 5 contains the string “abc”, its left child is the node number 4 and its right child is the node number 6. If the node does not have a child, or both, it would be shown by a -1 in the input. In other words, a leaf containing the string “aa” would be written as `aa -1 -1` in the input. The node numbers are 0-based and the 0th node is the root of the tree. For example take a look at the tree below. The numbers below the nodes show the corresponding line of input for the nodes.



```
9
a 1 3
b 2 8
d 6 7
c -1 5
i -1 -1
f 4 -1
g -1 -1
h -1 -1
e -1 -1
```

Output format.

The output consists of three line. The first line should contain the postorder traversal of the tree. In the traversal, each node's data string is printed followed by a space. The second line should contain the inorder traversal, and the last line should contain the preorder traversal of the tree. For example, the correct output for the given sample input is as follows.

```
g h d e b i f c a  
g d h b e a c i f  
a b d g h e c f i
```

Problem 2. Levelorder Traversal

Similar to problem 1 your program should read a binary tree from the input and print the levelorder traversal to the output. The input format is the same as problem 1, and the only difference in the output format is that in this problem you only need to print the levelorder traversal, and so the output would only have one line.

Example:

Input.

```
9
a 1 3
b 2 8
d 6 7
c -1 5
i -1 -1
f 4 -1
g -1 -1
h -1 -1
e -1 -1
```

Output.

```
a b c d e f g h i
```

Problem 3. Tree Reconstruction

In this problem you are going to reconstruct a tree using its postorder and inorder traversals, and print the level order traversal for the reconstructed tree.

Input format.

On the first line, you can read the number of elements in the tree. In the second line you can read the postorder traversal of the tree, in a format similar to the output format for problem 1, i.e. the data for each node is printed followed by a space. On the third line you can read the inorder traversal in a similar format. Example:

```
9
g h d e b i f c a
g d h b e a c i f
```

Output format.

The output should be one line containing the levelorder in a similar format to the problem 2.

```
a b c d e f g h i
```