

CS2100 Computer Programming Lab

Assignment 7

Trees and Recursion.

Date: Sept 28, 2015
To be submitted by 4th October.

The exercises in this lab are based on binary trees and recursion. Binary trees are inherently recursive and therefore several operations on trees can be very conveniently done using recursion. In this lab we will look at constructing binary trees, traversing them, and computing the diameter of a binary tree.

Problem 1 – Input format and creating the tree

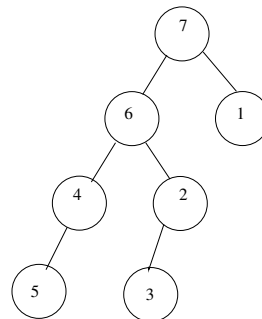
Recall the recursive definition of a binary tree. A null node is a binary tree. Using this as a base case, a binary tree can be recursively defined as a root node with two children left and right, both of which are binary trees. Treat this problem as a warm up for the remaining two problems. This problem asks you to create a binary tree whose specification is given in a file. Assume that the node-id (which is an integer) is the only data that is present in the node of a binary tree.

Input format

The input is taken from a text file. The first line of the file contains an integer N — number of nodes in the tree ($0 < N \leq 10000$). Next $N - 1$ lines contain $N - 1$ edges of that tree — Each line contains a pair (u, v) and either one of L or R separated by a space. This means that the u is the parent of node v . Furthermore, if the third parameter is L then v is the left child of u , else if the third parameter is R, then v is the right child of u . Since it is a binary tree, it is ensured that the value of u is not same for more than two different v 's. Hence, it can be assumed that the input describes a valid binary tree.

For example if the input is the following then the tree is as shown below:

```
7
7    6    L
2    3    L
4    5    L
6    2    R
7    1    R
6    4    L
```



To represent your tree you should use the following node structure:

```
struct node
{
    int data;
    struct node* left;
    struct node* right;
};
```

Implement a function called **create_tree** with the following signature:

struct node* create_tree(char* filename): The function takes as an input a file as given in the above format and returns the pointer to the root node representing the tree given in the file. This function should be used the following two problems as well.

Problem 2 – Tree traversals

Given a binary tree, there are various ways to traverse it. In this problem, you have to implement three traversals, namely Preorder, Inorder, and Postorder using **recursion**. Each of these is some ordering of elements present in the tree.

1. Preorder

- Display the root.
- Traverse the left subtree.
- Traverse the right subtree.

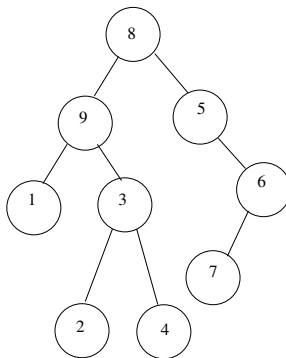
2. Inorder

- Traverse the left subtree.
- Display the root.
- Traverse the right subtree.

3. Postorder

- Traverse the left subtree.
- Traverse the right subtree.
- Display the root.

Assume that the description of the tree is given to you in the format as specified earlier. **Example :** The tree is given and the traversals are listed beside it.



Preorder traversal : 8 9 1 3 2 4 5 6 7

Inorder traversal : 1 9 2 3 4 8 7 6 5

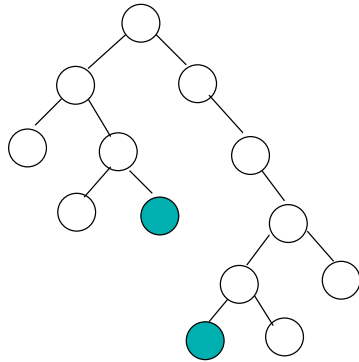
Postorder traversal : 1 2 4 3 9 7 6 5 8

Write a program that accepts a file as an input and prints all the 3 orders separated by a newline. Your output should not contain the words “Preorder traversal” etc. For each of the traversals in turn, separate the values using a space as shown above. For this question, you will have to implement code for three recursive functions whose signatures are given below. In each case the function prints the appropriate order traversal of the tree to the standard output.

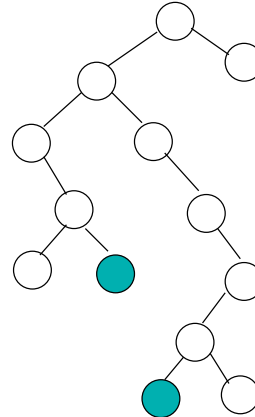
```
void preorder(struct node* curr);
void inorder(struct node* curr);
void postorder(struct node* curr);
```

Problem 3 – Diameter of a binary tree

The diameter of a tree (sometimes called the width) is the number of nodes on the longest path between two leaves in the tree. The diagram below shows two trees each with diameter 8, the leaves that form the ends of a longest path are shaded.



Diameter = 8 via root node.



Diameter = 8 not via root.

You are given a binary tree in the input format as specified earlier. Write a program to output the diameter of the tree. The diameter is the length of the longest path the tree is the number of edges we traverse from source to destination. You must implement a recursive function which computes the diameter of the tree and returns the diameter. Your program should accept a file which contains the tree description and outputs a number (which is the diameter). Sample input/outputs are given below.

Sample input 1

$$\begin{array}{ccc} 3 & & \\ 1 & 2 & L \\ 2 & 3 & R \end{array}$$

Sample output 1

2

Sample input 2

5		
1	2	L
2	3	L
4	5	R
3	4	L

Sample output 2

4

Submission Guidelines

Please follow the submission guidelines as usual. Create a roll-number-Lab7.tar.gz tar ball containing the 3 different files appropriately named.