CS5403 Data Structures and Algorithms

Assignment 5

Assigned date: October 29, Thursday

Due date: November 12, Thursday

Do not use a separate header file, application file and implementation file. Instead, put all your code in a single file called hw6.cpp and submit it via the Blackboard website.

Please include the return 0; statement in main() in this assignment and all future assignments.

Tasks:

In this assignment you will implement all three binary-tree traversal methods. More specifically, you will implement 3 recursive functions for in-order, pre-order and post-order traversal (one for each). Then you will implement a main function which, interacts with the user and utilize the traversal functions that you implemented. Basically, your main function should allow a user to input the contents of a binary tree obtained by one of the traversal methods. For example, user can type:

C PRE 1 3 4 5 7 3 2 4 5

To ask your program to create a binary tree which contains the provided integers, and it's pre-order traversal output is exactly the given order, 1 3 4 5 7 3 2 4 5. (here C in the command means create)

Similarly if user enters C IN 1 3 4 5 7 3 2 4 5, your program should create a binary tree containing the provided integers, and the tree's in-order traversal output should be exactly 1 3 4 5 7 3 2 4 5. Notice that, although it contains same set of integers, this tree would be structurally different than previous example.

Notice that there may be multiple binary-tree constructions, which lead to the same in-order traversal. You just need to use one. One possible way of doing this is to create a binary tree with the number of nodes is equal to the number of elements given in the input. Then you can do the proper traversal on the empty tree, and appropriately overwriting the contents one by one as you go.

Once the tree is constructed, your main function further interacts with the user to print out the tree as desired. For instance if user types:

P POST

your program should do post-order traversal of the tree and print the output. In summary your program should support the following commands.

C PRE 1 2 3 ….
C POST 1 2 3….
C IN 1 2 3 ….
P PRE
P POST
P IN