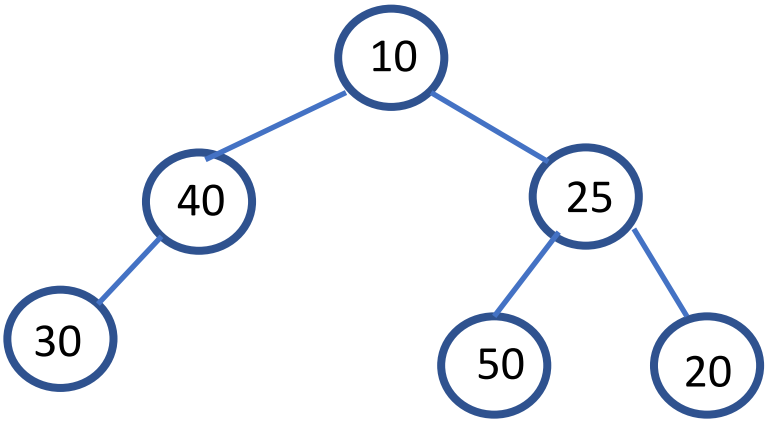
COMP 2611 – Data Structures

Lab 4

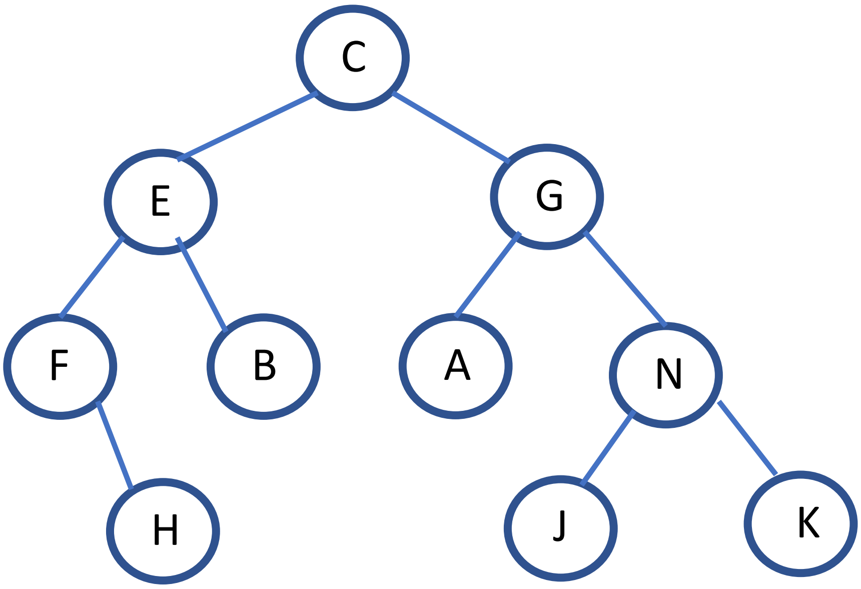
Binary Trees

(1) Give the preorder, inorder, and postorder traversals of the following binary trees:

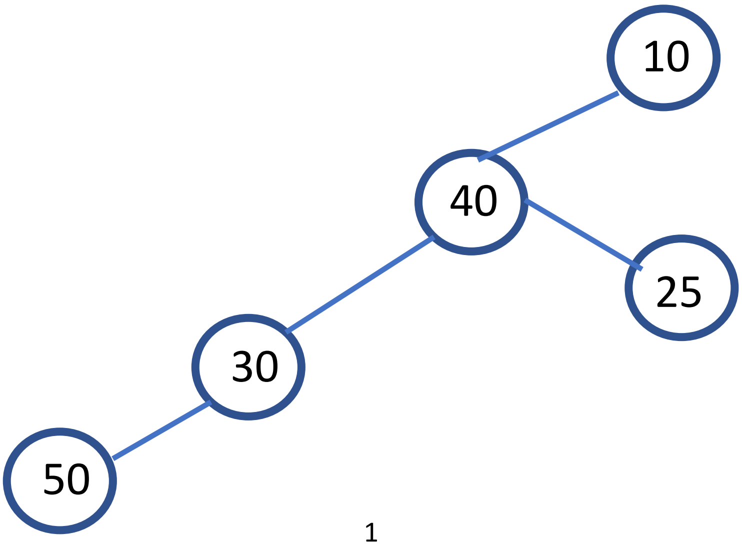
(a)



(b)



(c)



(2) A node in a binary tree can be declared as follows:

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

The source file, `BinaryTree.cpp`, contains the declaration for *BTNode*.

(a) Write the code for the *createBTNode* function with following prototype:

```
BTNode * createBTNode (int n);
```

(b) Wrte the code for the *preOrder*, *inOrder*, and *postOrder* functions with the following prototypes:

```
void preOrder (BTNode * root);
void inOrder (BTNode * root);
void postOrder (BTNode * root);
```

The functions must all be recursive and should simply display the value stored in the node when it is "visited".

(c) In the *main* function of `BinaryTree.cpp`, write code to create each of the six nodes of the binary tree shown in Question 1(a).

(d) Connect the six nodes as shown in Question 1(a) and set the value of *root* accordingly.

(e) Call the *preOrder*, *inOrder*, and *postOrder* functions with the value of *root* and ensure that the results obtained correspond with the answer for Question 1(a).

(3) (a) The *moment* of a tree is the number of nodes in the tree. Write a recursive function, *moment*, with the following prototype, which returns the moment of the binary tree:

```
int moment (BTNode * root);
```

(b) Write a recursive function, *numOneChild*, with the following prototype, which returns the amount of nodes in the binary tree that have exactly one child:

```
int numOneChild (BTNode * root);
```

(c) Write a recursive function, *numNonTerminal*, with the following prototype, which returns the amount of non-terminal nodes in the binary tree:

```
int numNonTerminal (BTNode * root);
```

A non-terminal node is any node that is not a leaf.

(4) Write a non-recursive algorithm to perform an *inOrder* traversal of a binary tree.