COMP 203 Lab 8

Tree and Queue

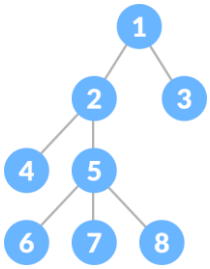
Implement Tree abstract data structure using Node data structure. Implement the following classes and methods: **(100pt)**

Keep in mind that in a tree, a parent may have more than 2 children. That is why you should create children of as a Queue data structure. Basically, a Node has "E data, Queue<Node<E>> childrenList, Node<E> parent".

1. Implement Node<E> class for Tree with constructor with the parameter E data. **(5pt)**
2. You can use Queue built in data structure of Java. You will use Queue data structure to save children of a parent. **(5pt)**
   Hint: You can create childrenList as follows:
   Queue<Node<E>> childrenList =new LinkedList<>();
3. Implement Integer QTree<E> class with the class variables and with no parameter constructor. **(5pt)**
4. InsertNode(Node<E> root, E parentValue, E addedValue) that inserts the node with the given value *addedValue* as a child of the node that has the value *parentValue*. You may use add function (as a helper function) of built-in data structure in java while implementing insertNode. **(20pt)**
5. Size() that returns the number of nodes in the tree. **(15pt)**
6. isPresent(Node<E> root, E value) that checks if there is a node with the given *value* in the given tree. **(20pt)**
7. Test your methods in the main by creating the following tree with the integer data type. (Creating the given tree: **15 pt**, testing 3 methods: **15 pt**)

Include comments of your code for each method and class.
Submit QTree.java to Canvas as a single java file. No other file types will be accepted.

Hint: You may use
import java.util.LinkedList;
import java.util.Queue;" to use Queue in your code.


1
2
3
4
5
6
7
8