BIL 212:Summer 2021

Lab 5

Download Tree.java, BinaryTree.java, AbstractTree.java, AbstractBinaryTree.java, LinkedBinaryTree.java and any other needed classes to compile these (like ArrayList.java) from Piazza site under Resources. Study theses classes. Implement the methods described below.

1. Write a driver that will first create a BinaryTree of integers. Insert integers in this tree using addRoot, addLeft, addRight, attach methods. Print the contents of the tree using the preorder() and postorder() methods.

2. Add to the `LinkedBinaryTree` class given in class a method
`int countDescendants(Position p)` which would return the count of the descendants of a position `p` including `p` itself. Your method should be recursive. Test your method by using the tree you constructed in Part 1.

3. Now, add to the `LinkedBinaryTree` class a method
`boolean contains(E target)` that will return true if the tree contains the data item **target**. Otherwise, it should return false. Use recursion. Do not call the preorder(), postorder() or positions() methods. Test your method.

4. In a driver class, write a method
`int minimum(BinaryTree<Integer> T))` which would take a BinaryTree of Integers as input and return the minimum integer in the tree. Use recursion! Do not call the preorder(), postorder() or positions() methods. Test your method.

5. In a driver class, write a method
`int sum(BinaryTree<Integer> T))` which would take a BinaryTree of Integers as input and return the sum of the integers in the tree. Use recursion! Do not call the preorder(), postorder() or positions() methods. Test your method.