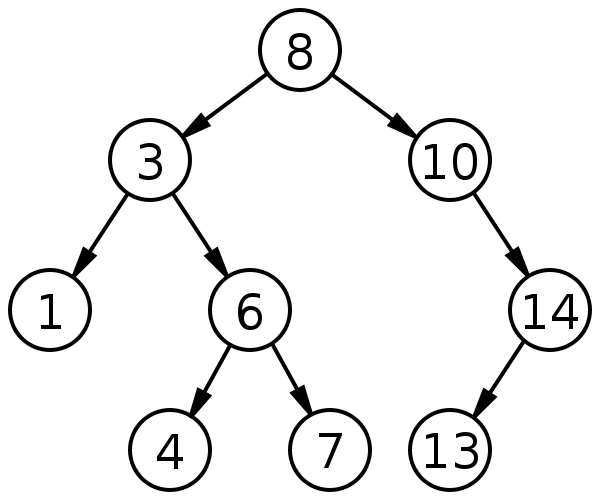
**COMP232 – Data Structures & Problem Solving**

**Fall 2020**

**Homework #4**

**Binary Trees**

1. Answer the questions below with respect to the following binary tree using the definitions from the text:



a. What value is at the root of the tree?

b. What values are in leaves of the tree?

c. What values are in internal nodes of the tree?

d. What values are descendants of 3?

e. What values are ancestors of 7?

f. What is the depth of node 6?

g. What is the height of the tree?

h. What is the path length from 8 to 4?

i. List the node values in the order they would be visited by:

i. An inorder traversal.

ii. A preorder traversal.

iii. A postorder traversal.

iv. A levelorder traversal.

2. This question refers to the BTNode<K, V> nested class declared in the CS232LinkedBinaryTree<K, V> class from the hw04 package. Assuming the variable node refers to a BTNode and that the tree is always sufficiently large, give a statement that identifies the node’s:

a. leftmost grand child.

b. great grand parent.

c. left child of right sibling, assuming node is a left child.

3. Complete the implementation of the MinKeyFinder class in the hw04 package. The No3Tests class contains tests that you can use to check your implementation of this class.

4. Implement the main() method in the MinKeyFinder class in the hw04 package so that it creates a CS232LinkedBinaryTree with at least 10 nodes and uses a preorder traversal with your visitor to print the key associated with the minimum value in the tree. Hint: The add() method of the CS232LinkedBinaryTree class is not implemented yet. You will be doing that in a later question. Therefore, you need to initialize the nodes of the binary tree using the constructor that takes arrays of keys and values: CS232LinkedBinaryTree(K[] keys, V[] vals).

5. Complete the implementation of the visitInOrder method in the CS232LinkedBinaryTree class in the hw04 package. The No5Tests class contains tests that you can use to check your implementation of this method.

6. Complete the implementation of the four-argument constructor for the CS232LinkedBinaryTree class in your hw04 package. The No6Tests class contains tests that you can use to check your implementations of this constructor.

7. Complete the implementation of the contains method in the CS232LinkedBinaryTree class in the hw04 package. The No7Tests class contains tests that you can use to check your implementations of this method. For full credit, implement contains so that it does not rely on the get or getNodeFromSubtree methods. Instead, add a helper method subtreeContains that returns a boolean value. This will be similar to the getNodeFromSubtree method. Then call subtreeContains from contains. Yes, this introduces repeated and unnecessary code, but the point here is not producing the most effective class, but for you to practice writing methods like this.

8. Complete the implementation of the add method in the CS232LinkedBinaryTree class in the hw04 package. The No8Tests class contains tests that you can use to check your implementations of this method.

9. Complete the implementation of the countLeafNodes method in the CS232LinkedBinaryTree class in the hw04 package. The No9Tests class contains tests that you can use to check your implementations of this method.