Written Assignment Content

Top of Form

Chapter-8:

R-8.1 The following questions refer to the tree of Figure 8.3.

1. Which node is the root?
   1. **/user/rt/courses/**
2. What are the internal nodes?
3. How many descendants does node cs016/ have?
   1. **9**
4. How many ancestors does node cs016/ have?
   1. **1**
5. What are the siblings of node homeworks/?
   1. **grades and programs/**
6. Which nodes are in the subtree rooted at node projects/?
   1. **papers/, demos/, buylow, sellhigh, market**
7. What is the depth of node papers/?
   1. **3**
8. What is the height of the tree?
   1. **4**

A diagram of a computer system

Description automatically generated

R-8.4 What is the running time of a call to T. height2(p) when called on a position p distinct from the root of T? (See Code Fragment 8.5.)

A screenshot of a computer code

Description automatically generated

R-8.21 In what order are positions visited during a pre-order traversal of the tree of Figure 8.8?

A diagram of a tree

Description automatically generated

**3, +, 1, x, 3, /, +, 9, -, 5, 2, -, 3, x, 7, -, 4, +, 4**

C-8.42 Describe how to clone a LinkedBinaryTree instance representing a (not necessarily proper) binary tree, with use of the add left and add right methods.

**Start at the root. Add the left and right nodes from the original tree and then recursively go down the tree left and right adding the left and right nodes as needed from the original tree until both left and rides nodes end up equaling None.**

Chapter-9:

R-9.3 What does each remove min call return within the following sequence of priority queue ADT methods: add(5,A), add(4,B), add(7,F), add(1,D), remove min( ), add(3,J), add(6,L), remove min( ), remove min( ), add(8,G), remove min( ), add(2,H), remove min( ), remove min( )?

R-9.5 The min method for the UnsortedPriorityQueue class executes in O(n) time, as analyzed in Table 9.2. Give a simple modification to the class so that min runs in O(1) time. Explain any necessary modifications to other methods of the class.

R-9.21 Show all the steps of the algorithm for removing the entry (16,X) from the heap of Figure 9.1, assuming the entry had been identified with a locator

C-9.26 Show how to implement the stack ADT using only a priority queue and one additional integer instance variable. *(Assume you have a PriorityQueue() that will create an object with two methods .put(key, value) (unsorted order) and .get() (removes and returns minimum element according to key) Note: Use the given approach and given methods for implementation, different methods or class name, would be penalised)*

Bottom of Form