Written Assignment Content

Top of Form

Chapter-8:

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

a. Which node is the root?

b. What are the internal nodes?

c. How many descendants does node cs016/ have?

d. How many ancestors does node cs016/ have?

e. What are the siblings of node homeworks/?

f. Which nodes are in the subtree rooted at node projects/?

g. What is the depth of node papers/?

h. What is the height of the tree?

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.)

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

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.

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