Data Structures: 505 22240 / ESOE 2012

Computer Assignment 5

Binary Search Tree

Due: three weeks from announcement by 9:00 p.m.

*Hand in online via **NTU COOL***

Total score: 200

This homework will familiarize you with the fields and methods of the **BinaryTree** and **BinaryTreeNode** classes. BinaryTree is an implementation of the Dictionary interface (which you also used in CA 4) as a binary search tree. Your task is to implement the operations of a binary search tree in the file **BinaryTree.cpp**.

Part I: Finding an Element in a Binary Search Tree (50%)

Complete the implementation of the find() method in BinaryTree.cpp by filling in the body of findHelper(). find() takes a key as its single parameter, and returns an element associated with that key, or null if there is none. (If there are several elements associated with a key, it doesn't matter which one is returned.) findHelper() helps by recursively finding and returning a node that contains the key (or null if no such node exists).

Take a look at insertHelper() for inspiration. find() should run in $O(d)$ time on a tree with depth d.

Part II: Removing an Element with a Given Key (120%)

Fill in the body of remove() method in BinaryTree.cpp. remove() takes a key as its single parameter, and removes one item having that key if the tree contains one. (If there are several elements associated with a key, it doesn't matter which one is removed.) remove() should not call find(). However, remove() SHOULD use the findHelper() method you wrote for Part I. Make sure remove() works for nodes that have *no* child or *one* child as well as *two* children.

remove() should run in $O(d)$ time if the depth of the tree is d.

Part III: Removing all Elements (30%)

There is a makeEmpty() method, which removes every node from a binary search tree and is also used in the constructor. Make sure that the memory of all existing nodes is properly released via *delete* to avoid memory leak.

You are free to add your own methods but do NOT change the prototype of the given methods. **TestBinaryTree.cpp** provides some test codes for the find() and remove() functions. You are welcome to modify the test codes and add further tests of your own as you please.