Data Structures- Fall, 2022 Prof. Rahouti

Homework 4

100 points

Consider the following skeleton code for a Binary Search Tree (BST) class:

```
typedef char ItemType;
struct TreeNode;
class TreeType
{
public:
  TreeType();
 ~TreeType();
  TreeType(const TreeType& originalTree);
  void MakeEmpty();
  bool IsEmpty() const;
  bool IsFull() const;
  int GetLength() const;
  ItemType GetItem(ItemType item, bool& found);
  void PutItem(ItemType item);
  void DeleteItem(ItemType item);
  void Print(std::ofstream& outFile) const;
private:
  TreeNode* root;
};
```

Where the struct is defined as follows:

```
struct TreeNode
{
  ItemType info;
  TreeNode* left;
  TreeNode* right;
};
```

Problem 1 (30 pts):

Implement the following member functions in the .cpp file in accordance with the given declarations in the skeleton code:

a. Constructor //3 pts
b. Destructor //3 pts
c. GetLength //5 pts
d. IsEmpty() //3 pts
e. IsFull() //3 pts
f. GetItem //7 pts
g. Print //6 pts

Problem 2: (18 pts)

Implement a member function IsBST to test whether the tree is a binary search tree, i.e., whether for each node in the tree, its value is larger than values stored in its left subtree, and smaller than values stored in nodes in its right subtree.

Problem 3: (18 pts)

Add a member function ***DeleteRoot*** to delete the root of the BST. Note that the BST property must be preserved after deletion.

Problem 4: (16 pts)

Add a member function ***DeleteSmallest*** that uses **recursion** to delete the value of the smallest item stored in the tree.

Problem 5: (18 pts)

Write a member function ***GetNodes*** that returns an array of items stored in the leaf nodes of the BST.