Assignment 5 : Binary Search Tree

Deadline: 08/01/2021 11:55 PM

Instruction:

1. Write your code in the c file named “BST.cpp”.
2. Avoid plagiarism. If you are found to adopt any unfair means you will get a straight 0.
3. Upload your code (only the cpp file) in elms.
4. Deadline is 08/01/2021 11:55 PM.

Task:

In this assignment you will have to implement the basic Set operations (Union, Intersection and Minus) using Binary Search Trees.

In “BST.cpp” file you will find all the necessary operations implemented under the class BST to operate a Binary Search Tree. The operations are listed below.

a) Search
b) Insert
c) Remove
d) Print Tree
e) Print in order
f) List all item

You will also find class Queue, which has all the necessary operations implemented to Operate a Queue.

Important: In the class we built a Binary Search Tree which could operate with integer type data. But in the assignment you will work with string type data. However, you don’t need to worry about that. The code in “BST.cpp” is made ready to accommodate string type data.

And to make things easier I have already implemented the Union operation. Your job is to implement Intersection and Minus operation using rules and logic of Set theory you learned in Mathematics. I have also discussed it in previous lecture. You can also refer to that. The function prototypes are written. You are advised to follow them strictly.

In the zip file you will also find another file named “AVL_tree.cpp”. This is also a binary search tree. But where a trivial Binary Search Tree is not automatically height balanced, an AVL tree can balance it’s height by itself. You can copy and paste your implementation of Intersection and Minus operation into “AVL_tree.cpp” file to see the difference. However you don’t need to submit this file.