CSE 2383: Data Structures and Algorithm Analysis

Challenge 3 – BST + U = BFF

Submission Window Opens:
Due Date Posted to Canvas

Points Available:
90 points for a working demonstration
10 points for correct submission & understandable code

Objectives:

- Demonstrate a working BST
- Demonstrate sufficient knowledge to implement a completed BST remove function

Assignment:

For this challenge, you must write a program that implements a BST derived from the code provided in class. There are two stages of code development (additional details are below): code completed prior to your demonstration and code completed during your demonstration. A series of function calls and their subsequent output are included below. You must demonstrate a working BST to the TA. As part of the demonstration process, the TA will also ask you to remove a section of your delete function and reproduce it from memory. Practice and study the implementation this data structure.

Prior to demonstrating your code:

1. Implement the BST code shown in the slides presented in class. You may omit the "destroy" function shown in the example BST class definition.
2. Write a "degree" function for your node class as described in lecture. The example code will not function without degree function.
3. Write a "find" function that will search your BST for a given value. The function must return a Boolean value of true if the value is found and a Boolean value of false if it is not found.
4. You will need to write the "public facing" versions of each function. This is fairly straightforward as the public facing functions simply call their corresponding private functions with the root node as a parameter. Example:

```
bool BST::insert(int data)
{
```

```
        return insert(root, data);
    }
```

5. Write traversal functions for In-Order, Pre-Order, and Post-Order traversals as described in the slides. These functions must output a text label indicating which traversal method is being used (see example execution).

To be completed DURING YOUR DEMO:

1. You will need to remove, recall, and rewrite the code for your delete function beginning at the line in the example code that reads:
   `int deg = node->degree();`
   You will be asked to explain in English what each step of this code is doing.

As with the previous challenge, you must fully pass the demonstration before a grade will be assigned to your submission. Grades are assigned based on day of submission (see syllabus). Once you have demonstrated your submission, you must upload it to Canvas within 48 hours.

You will need to bring your laptop to the TA for demonstration. ***If you cannot bring your laptop, make arrangements ahead of time with the TA to demonstrate it some other way.*** Arrangements must be made at least a day ahead of your planned demonstration.

Deliverables

1. Demo your working code to the class TA ***before uploading*** it to Canvas. You cannot proceed to step 2 before doing this.
2. Once your code is working and you've demoed it to the TA, upload all your code to Canvas as a single ZIP file. Name your ZIP file <netID>_2383_Ch<challenge_number>.zip, where <netID> is your MSU Net ID and <challenge_number> is the number assigned to the Challenge.

Example Code:

```
BST b;
b.insert(56);
b.insert(34);
b.insert(70);
b.insert(5);
b.insert(1);
b.insert(42);
b.insert(40);
b.insert(52);
b.insert(62);
b.insert(57);
b.insert(89);
b.insert(90);

cout << b.find(44) << endl;
cout << b.find(62) << endl;

b.remove(62);
cout << b.find(62) << endl;

b.insert(44);
cout << b.find(44) << endl;

b.postOrder(cout);
b.preOrder(cout);
b.inOrder(cout);
```

Example Execution/Output:

```
0
1
0
1
postOrder: 1 5 40 44 52 42 34 57 90 89 70 56
preOrder: 56 34 5 1 42 40 52 44 70 57 89 90
inOrder: 1 5 34 40 42 44 52 56 57 70 89 90
```