Lab #11
Summer 2024

Requirements

In this lab, you will cover creating and maintaining a Binary Search Tree. Remember that you **are encouraged** to use helper functions. In this lab you are given the following partial struct definition:

```
// Must complete in lab11.c
typedef struct BST BST;
```

1.1 initBST

```
// O(1)
```

```
BST * initBST()
```

Info: This function initializes and returns a pointer to a BST struct on success, or NULL if initialization was not successful. Your grade for this function includes your BST implementation.

1.2 insertBST

```
// O(log(n))
```

```
int insertBST(BST *tree, int data)
```

Info: This function takes a BST and a data integer. It will insert the given data onto the tree and return 0 on success, or 1 if insertion was not successful. **You may assume no duplicate items will be inserted**.

1.3 getSizeBST

```
// O(1)
```

```
int getSizeBST(BST *tree)
```

Info: This function takes a BST and returns the number of elements in the BST.

1.4 getMinBST

```
// O(log(n))
```

```
int getMinBST(BST *tree)
```

Info: This function takes a BST and returns the minimum value in the tree. You may assume the tree is not empty.

1.5 postOrderPrintBST

```
// O(n)
```

```
void postOrderPrintBST(BST *tree)
```

Info: This function takes a BST and prints the elements in the BST in post-order. The lecture slides discussing this are on Canvas under Modules.

1.6 inOrderPrintBST

```
// O(n)

void inOrderPrintBST(BST *tree)
```

Info: This function takes a BST and prints the elements in the BST in in-order. The lecture slides discussing this are on Canvas under Modules.

1.7 freeBST

```
// O(n)

void freeBST(BST *tree)
```

Info: This function takes a BST and frees all memory allocated to it.

Description

To get started on this lab, type the following while logged in to hellbender.rnet.missouri.edu:

```
cs2050start lab11
cd lab11
make
./a.out
```

For the lab assignment, you are to start with the starter code provided in the lab11 directory. This starter code is an outline of what you need to do. This time, you will need to change lab11.c as well as lab11main.c. The lab11main.c file that is provided does some rudimentary testing, but you will probably want to do more. You only need to submit lab11.c.

Submission Information

Submit this assignment by the mucsmake command to submit lab11.c:

```
mucsmake 2050 lab11 lab11.c
```

Rubric: 35 points

1. Write required *initBST* function
   * 10 points
2. Write required *insertBST* function
   * 6 points
3. Write required *getSizeBST* function
   * 1 points
4. Write required *getMinBST* function
   * 3 points
5. Write required *postOrderPrintBST* function
   * 5 points
6. Write required *inOrderPrintBST* function
   * 5 points
7. Write required *freeBST* function
   * 5 points

Notice:

1. All of your lab submissions must include documentation in the form of code comments to receive full points. In addition, your program is expected to have **a comment header** at the top that includes your name, pawprint, the course you are taking, and the lab that you solved.
2. All of your lab submissions must compile under GCC using the −*Wall* and −*Werror* flags to be considered for a grade (note that the Makefile should take care of this for you in this case).
3. Do **NOT** change any files other than lab11.c and lab11main.c. Struct definitions and additional helper functions must be placed in your lab11.c file.