CMP_SC 3050 SP2023

ASSIGNMENT 4: IMPLEMENT BINARY SEARCH TREE

DESCRIPTION

You are to implement the functions that have been "stubbed out" in bst.c. You will also need to define data types in bst.h. Basically, you are to implement a all of the needed Binary search Tree (BST) functions such that main.c will work properly when linked to your bst.o.

You are given a main() program, a Makefile, and some starter code. All you need to do is complete the actual code for the functions in bst.c and make any changes needed to bst.h.

HOW TO GET STARTED

The starter code is publicly available on GitHub. You can clone it using git via the following command:

```
git clone https://github.com/JimRies1966/cs3050sp2023A4.git
```

I recommend you clone this code somewhere under your home directory on tc.rnet.missouri.edu. You are welcome to clone this and work on your code on any platform you like. However, you should be aware that submissions will **only** be evaluated by the TAs on tc.rnet.missouri.edu. If, for example, something works on your machine but doesn't compile on tc.rnet.missouri.edu, you will get a zero.

Once you have cloned things down, you should cd to the newly cloned directory and type "make". This will build the code and leave you with an executable file called "mybst".

NOTES

- You should not need to change any of the files except bst.c and bst.h. When you have completed the assignment, submit both bst.c and bst.h on Canvas. The TAs will clone fresh starter code down and copy in your bst.c and bst.h in order to evaluate it.
- You will probably want to uncomment and possibly add code to print things out in main() while you are working on the assignment, but you should make sure your code works with the original main.c before you submit it.

SAMPLE OUTPUT

```
jimr@jimrsurfacepro9:~/CS3050/SP2023/assignments/A4$ ./mybst
Creating an empty BST...

***In order:
Abbie Ries (111111)Maggie Durant (121212)Jim Ries (123456)Paul Durant
(212121)Katherine Durant (222222)Charlotte Ries (333333)Alex Durant (444444)Cisco
Ries (555555)Murphy Ries (666666)Allison Ries (777777)Laura Ries (789012)Larry Ries
(888888)Marie Ries (999999)

***Preorder:
Jim Ries (123456)Abbie Ries (111111)Maggie Durant (121212)Laura Ries
(789012)Charlotte Ries (333333)Paul Durant (212121)Katherine Durant (222222)Cisco
Ries (555555)Alex Durant (444444)Murphy Ries (666666)Allison Ries (777777)Larry Ries
(888888)Marie Ries (999999)

***Postorder:
Maggie Durant (121212)Abbie Ries (111111)Katherine Durant (222222)Paul Durant
(212121)Alex Durant (444444)Allison Ries (777777)Murphy Ries (666666)Cisco Ries
(555555)Charlotte Ries (333333)Marie Ries (999999)Larry Ries (888888)Laura Ries
(789012)Jim Ries (123456)


Looking for 111111
Found: Abbie Ries (111111)


Looking for 222222
Found: Katherine Durant (222222)

Deleting 111111 ...
Maggie Durant (121212)Jim Ries (123456)Paul Durant (212121)Katherine Durant
(222222)Charlotte Ries (333333)Alex Durant (444444)Cisco Ries (555555)Murphy Ries
(666666)Allison Ries (777777)Laura Ries (789012)Larry Ries (888888)Marie Ries
(999999)

Deleting 222222 ...
Maggie Durant (121212)Jim Ries (123456)Paul Durant (212121)Charlotte Ries
(333333)Alex Durant (444444)Cisco Ries (555555)Murphy Ries (666666)Allison Ries
(777777)Laura Ries (789012)Larry Ries (888888)Marie Ries (999999)

Adding 000000 ...

Adding 654321 ...
Ned Needleman (0)Maggie Durant (121212)Jim Ries (123456)Paul Durant (212121)Charlotte
Ries (333333)Alex Durant (444444)Cisco Ries (555555)Lou Reed (654321)Murphy Ries
(666666)Allison Ries (777777)Laura Ries (789012)Larry Ries (888888)Marie Ries
(999999)
```