Concerning the following functions:

```c
int findBST(bst *tree, void *value); //returns 1 if found, 0 otherwise
bstNode *findBSTNode(bst *tree, void *value); //returns 0 if not found
```

Are these essentially the same operation, only with different return types? If so, wouldn't (int) findBST be redundant, since you could just run findBSTNode and test for a NULL value?

Also, it seems like it would be a good strategy then to write findBSTNode first, then implement findBST as a call to findBSTNode that returns 0 if findBSTNode returns a NULL value and 1 otherwise.

Am I wrong?
cdyancey wrote on Sat, 25 February 2017 16:11

Concerning the following functions:

```c
int findBST(bst *tree, void *value); //returns 1 if found, 0 otherwise
bstNode *findBSTNode(bst *tree, void *value); //returns 0 if not found
```

Are these essentially the same operation, only with different return types? If so, wouldn't (int) findBST be redundant, since you could just run findBSTNode and test for a NULL value?

Also, it seems like it would be a good strategy then to write findBSTNode first, then implement findBST as a call to findBSTNode that returns 0 if findBSTNode returns a NULL value and 1 otherwise.

Am I wrong?

I noticed Dr. Lusth didn't try to correct this logic, so I'm assuming this is how it's supposed to go?

---

lusth wrote on Tue, 07 March 2017 01:00:39 GMT

My findBST calls findBSTNode.

---

bmbaker1 wrote on Tue, 07 March 2017 16:49:20 GMT

^^^ That is what mine does, but I ended up never using findBST(). I just used findBSTNode(). I have it implemented for the ol' file switcharoo, but I also have an 'unused function' warning...

---

bkaaron wrote on Wed, 08 March 2017 16:34:27 GMT

Woh, Woh, Woh, hold the phone here a second.

findBST is NOT the same as findBSTNode

findBST returns the frequency of a value in the tree, any positive int, or 0 if the value (in the passed node of course) was NOT found at all.
It DOES NOT function as a binary/boolean function of found or not found.

Project Description under INTRODUCTION section states:

"You will compare the performance of vanilla binary search trees versus red-black trees by reading in a corpus of text, storing the word and phrases therein into a search tree, and then performing operations on the resulting tree. The corpus will be stored in a file; the commands for manipulating the tree will be stored in a file as well. You will implement the following operations for both kinds of trees:
- insert which inserts a value into the tree (or updates a frequency count)
- delete which decreases a frequency count (or removes a value from the tree)
- find which reports the frequency of a value in the tree"

Lusth please respond to confirm!

bkaaron
I do not believe there is anyway to return the frequency of a node's phrase from bst.c without casting it to a vbst or rbt object first. I'm pretty sure that we are not supposed to do that. That being said, per my other reply in this thread, I never used findBST(), so...

If what you say is true, then we need a specs change. I quoted directly from the specs. It tells us what find is supposed to do. We need a response from Lusth before the official due date, but definitely before the grace period ends. This function, if tested as it is described in the spec, will fail out for anyone who returns boolean notation only.

I see my mistake. In regard to the vanilla findBST the spec is not incorrect if you return Boolean, because you handle the frequency return in the inheriting function's findRBT/findVBST respectively. This is my mistake. As long as those functions return non-boolean values you are fine. findBST need not return anything other than boolean 0/1.
You are correct Baker

Subject: Re: Find BST
Posted by lusth on Wed, 08 Mar 2017 19:21:01 GMT

Glad *that's* settled.