Where do the functions that we have written for comparing and displaying vbstValues and rbtValues go?
I would like to keep them in their respective .c files but since they are not part of the public interface, they cannot be used by any other class and they need to be used in bstrees.c when creating a new vbst or rbt.

Also, should the struct for vbstValue and rbtValue have display and compare function pointers? I saw it done that way in a post somewhere on here but I can't see why we would need to have them stored in each value.

---

I have them as private functions in vbst.c and rbt.c respectively.

They are used in bst.c... see the answer to your next questions...

The struct for vbstValue and rbtValue should have display and compare function pointers...

So... newVBST() will take compare and display functions such as 'compareString' and 'displayString', respectively...

The newVBSTValue() will take the display and compare functions you sent the new vbst tree.
The newBST() will take the displayVBSTValue and compareVBSTValue. This is created inside newVBST()

---

So like this?

typedef struct vbst
Then were do the displayVBSTValue() and compareVBSTValue come in and how are they accessed?

Subject: Re: Display and Compare functions
Posted by bmbaker1 on Sat, 04 Mar 2017 22:04:11 GMT

View Forum Message <> Reply to Message

Quote:Then were do the displayVBSTValue() and compareVBSTValue come in and how are they accessed?
They are used inside of bst.c, specifically when you are inserting (compareVBSTValue) and printing (displayVBSTValue). When you print from bst.c, you will have a bstnode *node.

node->value will be a vbstValue object. To print the information held in the vbstValue object you would use
bstree->display(stdout, node->value);

The same for comparing vbstValue's inside of bst.c
bstree->compare(nodeA->value, nodeB->value);