Subject: Implementing Basic BST Structure Frequency Count
Posted by davidmccoy on Thu, 23 Feb 2017 00:07:25 GMT

In the provided bst.h from the spec sheet, a bstNode does not have a frequency count. Will a basic bst then merely move to the left child and then attempt insertion there? Moreover, when it comes time to implement vanillaBST, how would we not perform this incorrect behavior since now we'd want to increase frequency count instead of inserting?

Subject: Re: Implementing Basic BST Structure Frequency Count
Posted by josephmalafronte on Thu, 23 Feb 2017 00:50:54 GMT

I am also very confused by this. In our bst insert function what should we do if the value we are inserting is already in the tree.

Subject: Re: Implementing Basic BST Structure Frequency Count
Posted by Ben Tegtmeyer on Thu, 23 Feb 2017 01:25:33 GMT

According to the spec sheet all BST inserts are assumed to be unique. When inserting into a vanillaBST, check if the value already exists in the tree and either update the frequency count or insert the value based on if the value was found.

Subject: Re: Implementing Basic BST Structure Frequency Count
Posted by bmbaker1 on Thu, 23 Feb 2017 02:12:25 GMT

The way I am currently doing it, sending every value that I get to the bst... but when inserting in the bst module, I only check if its less than or greater than the current value... Never if its equal. That way it never inserts anything that is already in there.

Subject: Re: Implementing Basic BST Structure Frequency Count
Posted by lusth on Fri, 24 Feb 2017 15:24:25 GMT

^ That's OK, but your bst code should no nothing of frequency counts.
So when inserting into a vanilla or rb tree, we are to call findBSTNode first, then update frequency or call insertBST depending on result of findBSTNode?

^ Correct