I'm confused as to what `swapToLeafBSTNode` is supposed to actually do and how it relates to `pruneBSTNode`... Can anyone explain?

---

**Subject:** Re: Purpose of swapToLeafBSTNode?  
**Posted by:** luth  
**Posted on:** Fri, 24 Feb 2017 15:28:26 GMT

`swapToLeaf` is just a normal bst delete, except:

* no splicing out the node to be deleted if it has a single child  
* no pruning of the leaf when the value to be deleted arrives at a leaf

---

**Subject:** Re: Purpose of swapToLeafBSTNode?  
**Posted by:** jkmitchell2  
**Posted on:** Fri, 24 Feb 2017 23:41:12 GMT

So `swapToLeaf(x)` will always replace `x` with its predecessor, and in the bigger picture, `delete(x)` will always replace `x` with its predecessor regardless of how many children `x` has?

If `x` has no predecessor, then is it replaced by the root? That's what the text book version of predecessor seems to handle this case.

---

**Subject:** Re: Purpose of swapToLeafBSTNode?  
**Posted by:** jablanchard1  
**Posted on:** Sun, 26 Feb 2017 23:57:38 GMT

I found this to be a pretty useful sudo-code explanation of what is supposed to happen I think

---

**Subject:** Re: Purpose of swapToLeafBSTNode?  
**Posted by:** davidmccoy  
**Posted on:** Mon, 27 Feb 2017 01:26:18 GMT

I was struggling with this, good post!
@jkmitchell2, good find!

Now this brings the next question.. if there exists both a left and right subtree for a node, which should we swap with: predecessor or successor?

Subject: Re: Purpose of swapToLeafBSTNode?
Posted by josephmalafronte on Mon, 27 Feb 2017 04:40:35 GMT

"The swapToLeafBST function should prefer swapping with a predecessor over swapping with a successor."
From the instruction sheet

Subject: Re: Purpose of swapToLeafBSTNode?
Posted by nltollman on Mon, 27 Feb 2017 04:44:07 GMT

Good question... One solution might be since we are using RBT, which is a self-balancing tree, we see which swap would be better with the balancing factor part? swapping down the left subtree or the right subtree by seeing which way would result in the fewest rotations? As far as VBST, I'm not sure. But with RBT, I see it as a comparison on which subtree of node N produces the least rotations. Also, I just saw this in the assign2 instructions.

File Attachments
1) Capture.PNG, downloaded 135 times