










Delete a Node with Two Child in a Binary Search Tree

Removing nodes that have two children is the hardest case to implement. Removing a node like this produces two subtrees that are no longer connected to the original tree structure. How can we reconnect them? One method is to find the smallest value in the right subtree of the target node and replace the target node with this value. Selecting the replacement in this way ensures that it is greater than every node in the left subtree it becomes the new parent of but also less than every node in the right subtree it becomes the new parent of. Once this replacement is made the replacement node must be removed from the right subtree. Even this operation is tricky because the replacement may be a leaf or it may itself be the parent of a right subtree. If it is a leaf we must remove its parent's reference to it. Otherwise, it must be the right child of the target. In this case, we must replace the target value with the replacement value and make the target reference the replacement's right child.

Let's finish our `remove` method by handling the third case. We've provided some code again for the first two cases. Add some code now to handle target nodes with two children. Any edge cases to be aware of? What if the tree has only three nodes? Once you are finished this will complete our deletion operation for binary search trees. Nice job, this is a pretty hard problem!

	The <code>BinarySearchTree</code> data structure should exist.
	The binary search tree should have a method called <code>remove</code> .
	Trying to remove an element that does not exist should return <code>null</code> .
	If the root has no children, deleting it should set the root to <code>null</code> .
	The <code>remove</code> method should remove leaf nodes from the tree.
	The <code>remove</code> method should remove nodes with one child.

	Removing the root in a tree with two nodes should set the second to be the root.
	The <code>remove</code> method should remove nodes with two children while maintaining the binary search tree structure.
	The root should be removable on a tree of three nodes.