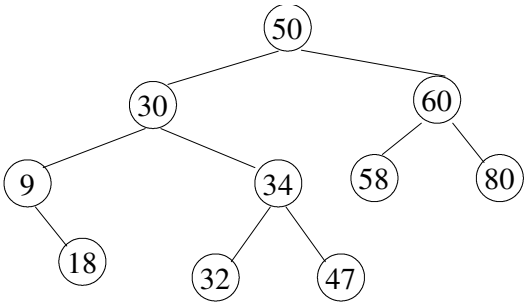
Data Structures (810:052) Name:_____

1. Complete the `findHelper` and `findReplacement` functions of the `remove` method for the BST class.

```
def remove(self, item):
    """Removes the item an returns data if item is found or None otherwise."""
    def findHelper(tree, parent):
```




```
    # end findHelper
    def findReplacement(tree, parent):

    # end findReplacement
    if self.isEmpty():
        return None
    elif self._tree.getRoot() == item and self._tree.getLeft().isEmpty():
        #
        #
        itemValue = self._tree.getRoot()
        self._tree = self._tree.getRight()
    elif self._tree.getRoot() == item and self._tree.getRight().isEmpty():
        #
        #
        itemValue = self._tree.getRoot()
        self._tree = self._tree.getLeft()
    else:
        #
        #
        itemValue, itemSubtree, itemParent = findHelper(self._tree, None)
        if itemValue == None:
            return None

        if itemSubtree.getLeft().isEmpty():
            if itemParent.getLeft() == itemSubtree:
                itemParent.setLeft(itemSubtree.getRight())
            else:
                itemParent.setRight(itemSubtree.getRight())

        elif itemSubtree.getRight().isEmpty():
            if itemParent.getLeft() == itemSubtree:
                itemParent.setLeft(itemSubtree.getLeft())
            else:
                itemParent.setRight(itemSubtree.getLeft())
        else:  #  item being removed has two children
            replacementValue,replacementSubtree,replacementParent=findReplacement(itemSubtree.getLeft(),itemSubtree
            itemSubtree.setRoot(replacementValue)
            if replacementParent == itemSubtree:
                itemSubtree.setLeft(replacementSubtree.getLeft())
            else:
                replacementParent.setRight(replacementSubtree.getLeft())

    self._size -= 1
    return itemValue
```

2. Complete the comments in the above code.

3. The shape of a BST depends on the order in which values are added (and deleted).

a) What would be the shape of a BST if we start with an empty BST and insert the sequence of values:

70, 90, 80, 5, 30, 110, 95, 40, 100

b) If a BST contains n nodes and we start searching at the root, what would be the worst-case theta Θ( ) notation for a successful search? (Draw the shape of the BST leading to the worst-case search)

4. We could store a BST in an array like we did for a binary heap, e.g. root at index 0, node at index i having left child at index 2 * i + 1, and right child at index 2 * i + 2.

a) Draw the above BST (after inserting: 70, 90, 80, 5, 30, 110, 95, 40, 100) stored in an array (use blank unused slots)

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19

b) What would be the worst-case storage needed for a BST with n nodes?

5. a) If a BST contains n nodes, draw the shape of the BST leading to best, successful search in the worst case.

b) What is the worst-case theta Θ( ) notation for a successful search in this “best” shape BST?