Our Solution(s) Run Code

Solution 1 Solution 2

```python
# Copyright © 2020 AlgoExpert, LLC. All rights reserved.

class BST:
    def __init__(self, value):
        self.value = value
        self.left = None
        self.right = None

    # Average: O(log(n)) time | O(log(n)) space
    # Worst: O(n) time | O(n) space
    def insert(self, value):
        if value < self.value:
            if self.left is None:
                self.left = BST(value)
            else:
                self.left.insert(value)
        else:
            if self.right is None:
                self.right = BST(value)
            else:
                self.right.insert(value)
        return self

    # Average: O(log(n)) time | O(log(n)) space
    # Worst: O(n) time | O(n) space
    def contains(self, value):
        if value < self.value:
            if self.left is None:
                return False
            else:
                return self.left.contains(value)
        elif value > self.value:
            if self.right is None:
                return False
            else:
                return self.right.contains(value)
        else:
            return True

    # Average: O(log(n)) time | O(log(n)) space
    # Worst: O(n) time | O(n) space
    def remove(self, value, parent=None):
        if value < self.value:
            if self.left is not None:
                self.left.remove(value, self)
        elif value > self.value:
            if self.right is not None:
                self.right.remove(value, self)
        else:
            if self.left is not None and self.right is not None:
                self.value = self.right.getMinValue()
                self.right.remove(self.value, self)
            elif parent is None:
                if self.left is not None:
                    self.value = self.left.value
                    self.right = self.left.right
                    self.left = self.left.left
                elif self.right is not None:
                    self.value = self.right.value
                    self.left = self.right.left
                    self.right = self.right.right
                else:
                    # This is a single-node tree; do nothing.
                    pass
            elif parent.left == self:
                parent.left = self.left if self.left is not None else self.right
            elif parent.right == self:
                parent.right = self.left if self.left is not None else self.right
        return self

    def getMinValue(self):
        if self.left is None:
            return self.value
        else:
            return self.left.getMinValue()
```

Your Solutions Run Code

Solution 1 Solution 2 Solution 3

```python
# Do not edit the class below except for
# the insert, contains, and remove methods.
# Feel free to add new properties and methods
# to the class.
class BST:
    def __init__(self, value):
        self.value = value
        self.left = None
        self.right = None

    def insert(self, value):
        # Write your code here.
        # Do not edit the return statement of this method.
        return self

    def contains(self, value):
        # Write your code here.
        pass

    def remove(self, value):
        # Write your code here.
        # Do not edit the return statement of this method.
        return self

```

Custom Output Raw Output Submit Code

Run or submit code when you're ready.