Our Solution(s) Run Code

Solution 1

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

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


# O(n) time | O(d) space
def validateBst(tree):
    return validateBstHelper(tree, float("-inf"), float("inf"))


def validateBstHelper(tree, minValue, maxValue):
    if tree is None:
        return True
    if tree.value < minValue or tree.value >= maxValue:
        return False
    leftIsValid = validateBstHelper(tree.left, minValue, tree.va
    return leftIsValid and validateBstHelper(tree.right, tree.va

```

Your Solutions Run Code

Solution 1 Solution 2 Solution 3

```python
# This is an input class. Do not edit.
class BST:
    def __init__(self, value):
        self.value = value
        self.left = None
        self.right = None


def validateBst(tree):
    # Write your code here.
    pass

```

Our Tests Custom Output Submit Code

```
import unittest


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


bright = BinaryTree
if hasattr(program, "BST"):
    bright = program.BST


class BST(bright):
    def insert(self, value):
        if value < self.value:
            if self.left is None:
                self.left = BST(value)
            else:
```

**Run or submit code when you're ready.**