

Solution 1

Solution 2

```
1 # Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 class BST:
4     def __init__(self, value):
5         self.value = value
6         self.left = None
7         self.right = None
8
9     # Average: O(log(n)) time | O(log(n)) space
10    # Worst: O(n) time | O(n) space
11    def insert(self, value):
12        if value < self.value:
13            if self.left is None:
14                self.left = BST(value)
15            else:
16                self.left.insert(value)
17        else:
18            if self.right is None:
19                self.right = BST(value)
20            else:
21                self.right.insert(value)
22        return self
23
24    # Average: O(log(n)) time | O(log(n)) space
25    # Worst: O(n) time | O(n) space
26    def contains(self, value):
27        if value < self.value:
28            if self.left is None:
29                return False
30            else:
31                return self.left.contains(value)
32        elif value > self.value:
33            if self.right is None:
34                return False
35            else:
36                return self.right.contains(value)
37        else:
38            return True
39
40    # Average: O(log(n)) time | O(log(n)) space
41    # Worst: O(n) time | O(n) space
42    def remove(self, value, parent=None):
43        if value < self.value:
44            if self.left is not None:
45                self.left.remove(value, self)
46        elif value > self.value:
47            if self.right is not None:
48                self.right.remove(value, self)
49        else:
50            if self.left is not None and self.right is not None:
51                self.value = self.right.getMinValue()
52                self.right.remove(self.value, self)
53            elif parent is None:
54                if self.left is not None:
55                    self.value = self.left.value
56                    self.right = self.left.right
57                    self.left = self.left.left
58                elif self.right is not None:
59                    self.value = self.right.value
60                    self.left = self.right.left
61                    self.right = self.right.right
62            else:
63                # This is a single-node tree; do nothing.
64                pass
65            elif parent.left == self:
66                parent.left = self.left if self.left is not None else self.right
67            elif parent.right == self:
68                parent.right = self.left if self.left is not None else self.righ
69        return self
70
71    def getMinValue(self):
72        if self.left is None:
73            return self.value
74        else:
75            return self.left.getMinValue()
76
```

Solution 1

Solution 2

Solution 3

```
1 # Do not edit the class below except for
2 # the insert, contains, and remove methods.
3 # Feel free to add new properties and methods
4 # to the class.
5 class BST:
6     def __init__(self, value):
7         self.value = value
8         self.left = None
9         self.right = None
10
11    def insert(self, value):
12        # Write your code here.
13        # Do not edit the return statement of this method.
14        return self
15
16    def contains(self, value):
17        # Write your code here.
18        pass
19
20    def remove(self, value):
21        # Write your code here.
22        # Do not edit the return statement of this method.
23        return self
24
```

Custom Output

Raw Output

Submit Code

Run or submit code when you're ready.