

Our Solution(s)Run Code

Solution 1

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 class BST {
4   constructor(value) {
5     this.value = value;
6     this.left = null;
7     this.right = null;
8   }
9 }
10
11 // O(n) time | O(d) space
12 function validateBst(tree) {
13   return validateBstHelper(tree, -Infinity, Infinity);
14 }
15
16 function validateBstHelper(tree, minValue, maxValue) {
17   if (tree === null) return true;
18   if (tree.value < minValue || tree.value >= maxValue) return false;
19   const leftIsValid = validateBstHelper(tree.left, minValue, tree.value);
20   return leftIsValid && validateBstHelper(tree.right, tree.value, maxValue);
21 }
22
23 exports.BST = BST;
24 exports.validateBst = validateBst;
25
```

Your SolutionsRun Code

Solution 1Solution 2Solution 3

```
1 // This is an input class. Do not edit.
2 class BST {
3   constructor(value) {
4     this.value = value;
5     this.left = null;
6     this.right = null;
7   }
8 }
9
10 function validateBst(tree) {
11   // Write your code here.
12 }
13
14 // Do not edit the line below.
15 exports.BST = BST;
16 exports.validateBst = validateBst;
17
```

```

1  class Node:
2      def __init__(self, value):
3          self.value = value
4          self.left = None
5          self.right = None
6
7  def insert(root, value):
8      if root is None:
9          return Node(value)
10
11     if value < root.value:
12         root.left = insert(root.left, value)
13     else:
14         root.right = insert(root.right, value)
15
16     return root
17
18 def search(root, value):
19     if root is None:
20         return False
21
22     if root.value == value:
23         return True
24
25     if value < root.value:
26         return search(root.left, value)
27     else:
28         return search(root.right, value)
29
30 def main():
31     root = None
32     values = [5, 3, 7, 2, 4, 6, 8, 1]
33     for value in values:
34         root = insert(root, value)
35
36     print("Binary Search Tree Structure:")
37     def print_in_order(node):
38         if node:
39             print_in_order(node.left)
40             print(node.value)
41             print_in_order(node.right)
42
43     print_in_order(root)
44
45     print("\nSearching for values:")
46     for value in values:
47         result = search(root, value)
48         print(f"Value {value}: {'Found' if result else 'Not Found'}")
49
50 if __name__ == "__main__":
51     main()

```

Run or submit code when you're ready.