

Our Solution(s)Run Code

Solution 1Solution 2

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 package main
4
5 type BST struct {
6     Value int
7
8     Left *BST
9     Right *BST
10 }
11
12 // Average: O(log(n)) time | O(log(n)) space
13 // Worst: O(n) time | O(n) space
14 func (tree *BST) Insert(value int) *BST {
15     if value < tree.Value {
16         if tree.Left == nil {
17             tree.Left = &BST{value: value}
18         } else {
19             tree.Left.Insert(value)
20         }
21     } else {
22         if tree.Right == nil {
23             tree.Right = &BST{value: value}
24         } else {
25             tree.Right.Insert(value)
26         }
27     }
28     return tree
29 }
30
31 // Average: O(log(n)) time | O(log(n)) space
32 // Worst: O(n) time | O(n) space
33 func (tree *BST) Contains(value int) bool {
```

Your SolutionsRun Code

Solution 1Solution 2Solution 3

```
1 package main
2
3 // Do not edit the class below except for
4 // the insert, contains, and remove methods.
5 // Feel free to add new properties and methods
6 // to the class.
7 type BST struct {
8     Value int
9
10    Left *BST
11    Right *BST
12 }
13
14 func (tree *BST) Insert(value int) *BST {
15     // Write your code here.
16     // Do not edit the return statement of this method.
17     return tree
18 }
19
20 func (tree *BST) Contains(value int) bool {
21     // Write your code here.
22     return false
23 }
24
25 func (tree *BST) Remove(value int) *BST {
26     // Write your code here.
27     // Do not edit the return statement of this method.
28     return tree
29 }
30
```

```

10 return 0
11
12 #Write your code here
13
14
15
16
17 #Write your code here
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

```

Run or submit code when you're ready.