

Our Solution(s)

Run Code

Your Solutions

Run Code

Solution 1

```
1 # Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 class BinaryTreeNode:
4     def __init__(self, value, left=None, right=None):
5         self.value = value
6         self.left = left
7         self.right = right
8
9
10 # O(n) time | O(d) space - where n is the number of nodes in the Binary Tree and d is the depth of the tree
11 def rightSiblingTree(root):
12     mutate(root, None, None)
13     return root
14
15
16 def mutate(node, parent, isLeftChild):
17     if node is None:
18         return
19     left, right = node.left, node.right
20     mutate(left, node, True)
21     if parent is None:
22         node.right = None
23     elif isLeftChild:
24         node.right = parent.right
25     else:
26         if parent.right is None:
27             node.right = None
28         else:
29             node.right = parent.right.left
30     mutate(right, node, False)
31
```

Solution 1 Solution 2 Solution 3

```
1 # This is the class of the input root. Do not edit it.
2 class BinaryTreeNode:
3     def __init__(self, value, left=None, right=None):
4         self.value = value
5         self.left = left
6         self.right = right
7
8
9 def rightSiblingTree(root):
10     # Write your code here.
11     pass
12
```

Our Tests

Custom Output

Submit Code



```

10         answerList = program.registeringAnswerList()
11         answer = answerList.getAnswer(i)
12         answerList.addAnswer(answer, answerList)
13
14     def test_case_2(self):
15         test = QuestionTest(2, answerList)
16         answerList = program.registeringAnswerList()
17         answer = answerList.getAnswer(i)
18         answerList.addAnswer(answer, answerList)
19
20     def test_case_3(self):
21         test = QuestionTest(3, answerList)
22         answerList = program.registeringAnswerList()

```

Run or submit code when you're ready.