

Our Solution(s)

Run Code

Solution 1

```
1 # Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 class Node:
4     def __init__(self, name):
5         self.children = []
6         self.name = name
7
8     def addChild(self, name):
9         self.children.append(Node(name))
10        return self
11
12    # O(v + e) time | O(v) space
13    def depthFirstSearch(self, array):
14        array.append(self.name)
15        for child in self.children:
16            child.depthFirstSearch(array)
17        return array
18
```

Our Tests

```
1 # Test 1
2 # Node
3
4 node = Node('root')
5 node.addChild('left')
6 node.addChild('right')
7
8 # Test 2
9 # Node
10
11 node = Node('root')
12 node.addChild('left')
13 node.addChild('right')
14 node.addChild('left-left')
15 node.addChild('left-right')
16 node.addChild('right-left')
17 node.addChild('right-right')
```

Your Solutions

Run Code

Solution 1    Solution 2    Solution 3

```
1 # Do not edit the class below except
2 # for the depthFirstSearch method.
3 # Feel free to add new properties
4 # and methods to the class.
5 class Node:
6     def __init__(self, name):
7         self.children = []
8         self.name = name
9
10    def addChild(self, name):
11        self.children.append(Node(name))
12        return self
13
14    def depthFirstSearch(self, array):
15        # Write your code here.
16        pass
17
```

Custom Output

Raw Output

Submit Code

Run or submit code when you're ready.