

Solution 1

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 class Program {
4     class Node {
5         let name: String
6         var children: [Node]
7
8         init(name: String) {
9             self.name = name
10            children = []
11        }
12
13        func addChild(name: String) -> Node {
14            let childNode = Node(name: name)
15            children.append(childNode)
16
17            return self
18        }
19
20        // O(v + e) time | O(v) space
21        func depthFirstSearch(array: inout [String]) -> [String] {
22            array.append(name)
23
24            for child in children {
25                child.depthFirstSearch(array: &array)
26            }
27
28            return array
29        }
30    }
31 }
32
```

Solution 1

Solution 2

Solution 3

```
1 class Program {
2     class Node {
3         let name: String
4         var children: [Node]
5
6         init(name: String) {
7             self.name = name
8             children = []
9         }
10
11        func addChild(name: String) -> Node {
12            let childNode = Node(name: name)
13            children.append(childNode)
14
15            return self
16        }
17
18        func depthFirstSearch(array: inout [String]) -> [String] {
19            // Write your code here.
20            return []
21        }
22    }
23 }
24
```

**Run or submit code when you're ready.**

---