AoL - Trees Quiz (Solutions)

[K/U] True or False

Write T or F beside each statement. For false statements, explain why they are false.

T or F	Statement	Explain if false
F	Figure 1 is a tree.	Trees cannot have cycles. G is a child of both D and E, which creates a cycle. Also H is not connected to the tree.
T	Figure 2 is a tree.	
F	Every node in a tree has exactly one parent.	The root of a tree does not have a parent.
F	In a tree, there can be multiple paths between 2 nodes.	If there were multiple paths, then it would create a cycle.

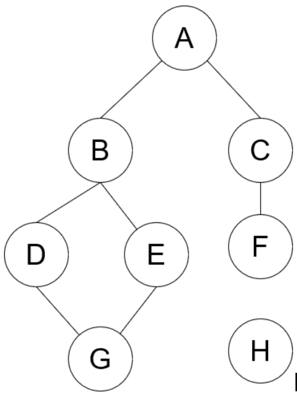


Figure 1

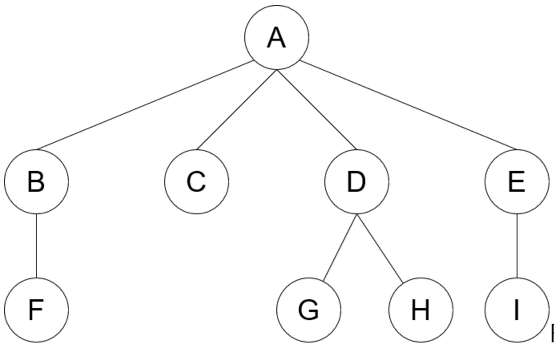
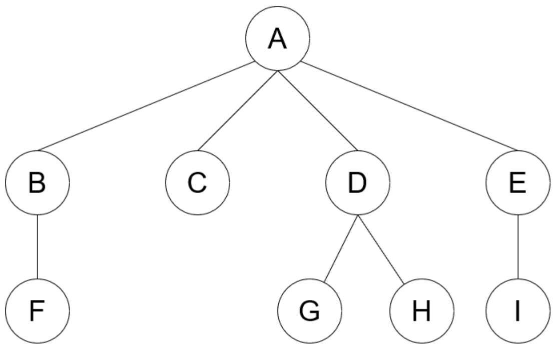


Figure 2

[COMM]

1. Explain the difference between pre-order and post-order traversal using an example.



Solution:

- Pre-order traversal means that the parent is visited before its children are visited.
- Post-order traversal means that the parent node is visited after its children are visited.
- Using this tree as an example, suppose we had a traversal algorithm that prints the value of the node when it is visited.
- Pre-order traversal would mean that nodes are visited in the following order: A,B,F,C,D,G,H,E,I. Notice node B is printed before its child F.
- Post-order traversal would mean that nodes are visited in the following order: F,B,C,G,H,D,I,E,A. Notice nodes G and H are visited before D, and the root A is visited last.

[K/U, T/I, A] Coding

1. Using the class definitions for the `Tree` and `Node` classes below, write a method `displaySum` for the `Tree` class to print the sum of all the nodes in the tree. Define any other methods you think are necessary.

```
class Node:
    def __init__(self, value):
        self.value = value
        self.children = []

    def addChild(self, value):
        self.children.append(Node(value))


class Tree:
    def __init__(self, value):
        self.root = Node(value)

    def displaySum():
        ...
```

Solution:

```
class Node:
    def addSum(self, sumOfChildren):
        sum = sumOfChildren
        for child in self.children:
            sum += child.addSum(sumOfChildren)
        sum += self.value
        return sum


class Tree:
    def displaySum(self):
        n = self.root.addSum(0) # start the sum at 0
        print("Sum of nodes is: ", n)
```

Look Fors:

- Wrapper method is used to call displaySum on the tree, but the recursive method is defined in Node class
- pre or post order traversal can be used
- print statement with context is provided