Feedback — Homework 3

Help Center

Thank you. Your submission for this homework was received.

You submitted this homework on **Thu 10 Mar 2016 2:45 PM PST**. You got a score of **80.00** out of **100.00**. You can attempt again, if you'd like.

"Give me six hours to chop down a tree and I will spend the first four sharpening the axe." - Abraham Lincoln

Question 1

Lists as recursive data structures

In Python, lists are primarily iterative data structures that are processed using loops. However, in other languages such as Lisp and Scheme, lists are treated primarily as recursive data structures and processed recursively.

This program defines a `NodeList` class that recursively creates a non-empty list of nodes. Note that a reference to the next node in the list is stored in the `_next` field of a `NodeList` object. What value is stored in the `_next` field to signal that there are no more nodes in the list?

Your Answer		Score	Explanation
○ False			
○ []			
○ 0			
◉ None	✔	10.00	Correct.
Total		10.00 / 10.00	

Question 2

Observe that the method `append` for the `NodeList` class is defined recursively. Given an existing list of n nodes (created as a `NodeList` object), how many calls to `append` are evaluated when appending a new node to the end of the list? Count only the calls performed during the process of appending of this $(n+1)$st node to the list, not those performed during the process of creating the original n-node input list.

Enter the answer as a math expression in n below.

You entered:

n

Preview Help

Your Answer		Score	Explanation
n	✔	10.00	Correct.
Total		10.00 / 10.00	

Question 3

Navigating trees

Our implementation of the `Tree` class includes a list of references to the subtrees associated with the root node. These references allow `Tree` methods to recursively traverse the tree downward from the root node to its children and then to its children's children and so on.

In some situations, the ability to traverse a tree in the opposite direction (upward) is also useful. This code defines a subclass `NavTree` (a navigable tree) of the class `Tree` that includes a `_parent` field. Methods in the `NavTree` class use this field to traverse from a node in a given tree to its parent node and so on. Note that in this representation, the root node of the given tree has value `None` in its `_parent` field to indicate that the node has no parent.

Based on the provided code for the `NavTree` class, which of the following implementations of `def get_root(self):` returns the root of the tree containing the subtree referenced by `self`?

Your Answer	Score	Explanation

○			

```
if self is None:
    return self
else:
    return self._parent.get_root()
```

◉	✔	10.00	Correct.

```
if self._parent is None:
    return self
else:
    return self._parent.get_root()
```

○			

```
return self._parent.get_root()
```

○			

```
if self._parent is None:
    return self
else:
    return get_root(self._parent)
```

Total		10.00 / 10.00	

Question 4

Given a tree, the *depth* of a node in that tree is the number of edges that connect the node to the root of the tree. In the provided code for the `NavTree` class, the nodes `"b"` and `"e"` have depths two and one, respectively, in the tree `tree_dcabe`.

Starting from this provided code for the `NavTree` class, which of the following implementations of `def depth(self):` returns the depth of the node referenced by `self`? Remember that the depth should be computed with respect to largest tree that contains `self`.

Your Answer	Score	Explanation
○		

```
return self._parent.depth() + 1
```

○		

```
if self._parent is None:
    return 0
else:
    return depth(self._parent)
```

◉	✔	10.00	Correct.

```
if self._parent is None:
    return 0
else:
    return self._parent.depth() + 1
```

○

```
if self._parent is None:
    return 0
else:
    return self._parent.depth()
```

Total	10.00 / 10.00

Question 5

As defined in the math notes on trees, a full binary tree is a tree in which each internal node has exactly two children. A full binary tree is *perfect* if every leaf in the tree has the same depth.

How many leaves does a perfect binary tree of height n have? Try some examples and look for a simple expression in n that reproduces the values generated by the examples. Enter the answer below as a math expression in n.

You entered:

```
2**n
```

Preview Help

Your Answer		Score	Explanation
2**n	✔	10.00	Correct.
Total		10.00 / 10.00	

Question 6

How many nodes (both internal and leaf) does a perfect binary tree of height n have?

To answer this question, you can either try some examples and look for pattern in n or convert your answer for question #5 into a sum and use the one of the formulas on this page

to reduce the sum to a single expression in n.

Enter the answer below as a math expression in n.

You entered:

```
2**(n+1)-1
```

Preview Help

Your Answer		Score	Explanation
2**(n+1)-1	✔	10.00	Correct.
Total		10.00 / 10.00	

Question 7

Applications to genealogy

Trees are extremely important tools for modeling genealogical data. One simple example is modeling the descendants of an important individual (let's call him "Luay XIV"). The root of the tree is labelled `"Luay XIV"` and its subtrees correspond to the descendants of the biological children of Luay XIV.

If Luay and his descendants can each have a maximum of four children and the height of Luay XIV's tree of descendants is three, what is the maximal number of descendants in Luay's tree? (Exclude Luay in this count.) Enter the answer below as a whole number.

You entered:

```
20
```

Your Answer		Score	Explanation
20	✖	0.00	
Total		0.00 / 10.00	

Question 8

Trees can also be used to model the ancestors of an individual (let's call him "Charles II"). In this model, the role of parent and child is reversed. The root of the tree is labelled `"Charles II"` and has two subtrees corresponding to the ancestors of Charles II's parents.

Note that ancestor trees are necessarily binary trees. However, the curious case of the real-life Charles II illustrates a complication of modeling genealogical data using trees. This image show the ancestor tree for Charles II going back several generations. (Charles is at the bottom.) Note that his family, the Hapsburgs, suffered from serious in-breeding. As a result, Charles II's family tree is not actually a tree, but more like a "family graph".

In the case of modeling ancestors, in-breeding violates the condition that every node in a tree (except for the root) should have a single parent node. In the case of Charles II's ancestor tree, several of Charles II's ancestors have multiple biological children who themselves are ancestors of Charles. (Remember that the parent-child relation is flipped in ancestor trees.)

Note our basic `Tree` class is fully capable of modeling Charles II's ancestors (even in the presence of in-breeding) since there is no check of the single-parent condition when an instance of the `Tree` class is created. However, the lack of this check will cause several of the methods for the `Tree` class to possibly return incorrect answers. Select the `Tree` methods below that may return an **incorrect** answer when an instance of the `Tree` class has multiple parents

Your Answer		Score	Explanation
☑ `num_nodes`	✔	3.00	
☐ `height`	✔	2.00	
☐ `children`	✔	2.00	
☑ `num_leaves`	✔	3.00	
Total		10.00 / 10.00	

Question Explanation

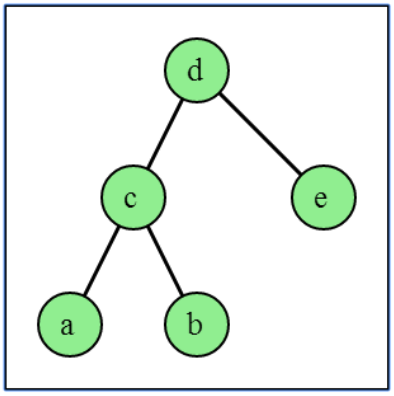
Remember to select those methods that may return an **incorrect** answer.

Question 9

Tree traversal

A *tree traversal* (or tree search) is a recursive method for visiting and processing all of the nodes in a tree. In the provided version of the `Tree` class, the `__str__` method performs a tree traversal to generate a string representation for a tree. In particular, this method creates a string representation for the root node's associated value and then appends the string representations for each subtree (which are recursively computed using the `str` method) to this string.

Consider a change to the `__str__` method in which line 32 is moved after the `for` loop in line 34-36 (and the commas are repositioned appropriately). In this modified version, the root's value is appended to the string representations for the root's sub-trees. Using this modified code, what is the string representation for the tree displayed below?



Your Answer		Score	Explanation
[[[a], [b], c], [e], d]			
[d, [c, [a], [b]], [e]]			
[[a, b, c], [e], d]			
[[[[a, b], c], e], [d]]	✖	0.00	Incorrect. Check the parentheses carefully.
Total		0.00 / 10.00	

Question 10

In the case of binary trees, the two subtrees of a node are usually referred to as the *left* subtree and the *right* subtree. For binary trees, there are three common types of tree traversals:

- **Pre-order traversals** - process root, process left subtree, process right subtree
- **Post-order traversals** - process left subtree, process right subtree, process root
- **In-order traversals** - process left subtree, process root, process right subtree

Examine the implementation of the `__str__` method for the `ArithmeticExpression` class in this

provided code. What type of tree traversal does the `__str__` method for the `ArithmeticExpression` class implement?

Your Answer		Score	Explanation
○ Pre-order traversal			
○ Post-order traversal			
◉ In-order traversal	✔	10.00	
○ None of the above			
Total		10.00 / 10.00	