



Assignment 4

Due Thursday, 9/27

September 24, 2018
CS: DS&A
PROOF SCHOOL

In this short assignment, we'll look at a couple issues related to the various recursive traverse functions we looked at in class.

Recall that we have three functions:

- In `traverse_1`, we just print out the keys as we reach them.
- In `traverse_2`, we pass in an empty list to the function call, and the function populates the list with the keys. We don't care what the function returns.
- In `traverse_3`, the user doesn't provide a list in the function call, but the function assembles the keys in an input list anyway, as in `traverse_2`. The final list is returned.

Question 1. Write a recursive function `traverse_4`, which does everything using return values. Its signature should be `def traverse_4(self, node="root")`, with no argument that is a list. Write it in the file `traversals_for_class.py`. (This should be fairly easy.)

Question 2. Recall from class that in `traverse_3`, we originally had the default argument `output=[]` in the function signature. This had the odd effect that when you ran the function twice, the second time it returned a list that had the keys twice! This is because in python, the default arguments are not created with each function call, but rather are created *once*, when the function is initially parsed, and then the same object is used (and modified) with each function call.

I said the standard thing to do was to use `output=None` as the default argument, and then in the body of the function, convert a `None` variable to an empty list. But ~~Jonathan~~ Josh pointed out that my function was behaving strangely: it missed some of the keys!

Take a look at the `traverse_3` code in `traversals_for_class.py`. It's not that hard to fix, but my question for you is: What in the world is going on? Why is this function misbehaving the way it is? (First, you'll have to investigate and figure out exactly how it's misbehaving.)

You can write your answer in a comment, and turn in `traversals_for_class.py` with the comment and the `traverse_4` code.