Deep Reverse

May 18, 2020

0.1 Problem Statement

Define a procedure, `deep_reverse`, that takes as input a list, and returns a new list that is the deep reverse of the input list.
This means it reverses all the elements in the list, and if any of those elements are lists themselves, reverses all the elements in the inner list, all the way down.

> Note: The procedure must not change the input list itself.

Example Input: `[1, 2, [3, 4, 5], 4, 5]` Output: `[5, 4, [5, 4, 3], 2, 1]`
Hint 1. Begin with a blank final list to be returned. 2. Traverse the given list in the reverse order. * If an item in the list is a list itself, call the same function. * Otheriwse, append the item to the final list.

0.1.1 Exercise - Write the function definition here

```
In [17]: def deep_reverse(arr):
             pass
```

Show Solution

0.1.2 Test - Let's test your function

```
In [16]: def test_function(test_case):
             arr = test_case[0]
             solution = test_case[1]

             output = deep_reverse(arr)
             if output == solution:
                 print("Pass")
             else:
                 print("False")
```

```
In [ ]: arr = [1, 2, 3, 4, 5]
        solution = [5, 4, 3, 2, 1]
        test_case = [arr, solution]
        test_function(test_case)
```

```
In [ ]: arr = [1, 2, [3, 4, 5], 4, 5]
        solution = [5, 4, [5, 4, 3], 2, 1]
        test_case = [arr, solution]
        test_function(test_case)
```

```
In [ ]: arr = [1, [2, 3, [4, [5, 6]]]]
        solution = [[[[6, 5], 4], 3, 2], 1]
        test_case = [arr, solution]
        test_function(test_case)
```

```
In [ ]: arr =  [1, [2,3], 4, [5,6]]
        solution = [ [6,5], 4, [3, 2], 1]
        test_case = [arr, solution]
        test_function(test_case)
```