UNOFFICIAL QUIZ *for* PRACTICE SOLUTIONS

QUIZ 5

Computer Science 61A . October 1, 2015 . alvinwan.com/cs61a

This quiz will not count towards your grade. It exists to simply gauge your understanding. You will have 5 minutes to complete this quiz. In that timespan, your goal is to complete one question and at least attempt the other two.

01. TREES

The following function takes a tree, and prints each "level" of the tree in sequential order, as a series of lists. Assume that you have access to the following functions:

- branches(t): *retrieves all branches of a tree, gives an iterable of all subtrees*
- root(t): *retrieves value of a tree*

```
def print_by_level(t):
    “”””
    >>> t = tree(1, [tree(3, [tree(2), tree(4)]), tree(6, [tree(5), tree(7)])])
    >>> print_by_level(t)
    [1]
    [3, 6]
    [2, 4, 5, 7]
    “”””

    lst, nxt, curr = [t], [], []
    while lst:
        for t in lst:
            nxt += branches(t)
            curr.append(root(t))
        print(curr)
        lst, nxt, curr = nxt, [], []
```

BONUS: After completing 03. below, implement `print_reverse_by_level`, which prints the tree's levels, but in reverse order.

```
>>> print_reverse_by_level(t)
[2, 4, 5, 7]
[3, 6]
[1]
```

02. TREES GALORE

```
def print_level_order(t):
    """
    >>> t = tree(1, [tree(3, [tree(2), tree(4)]), tree(6, [tree(5), tree(7)])])
    >>> print_level_order(t)
    1
    3
    2
    4
    6
    5
    7
    """

    def helper(t):
        children = branches(t)
        print(root(t))
        [helper(b) for b in children]

    helper(t)
```

03. REVERSING TREES

```
def print_reverse_level_order(t):
    """
    >>> t = tree(1, [tree(3, [tree(2), tree(4)]), tree(6, [tree(5), tree(7)])])
    >>> print_reverse_level_order(t)
    7
    5
    6
    4
    2
    3
    1
    """

    def helper(t):
        children = branches(t)
        [helper(b) for b in children[::-1]]
        print(root(t))

    helper(t)
```

BONUS

```
def print_reverse_by_level(t):
    “””
    >>> t = tree(1, [tree(3, [tree(2), tree(4)]), tree(6, [tree(5), tree(7)])])
    >>> print_reverse_by_level(t)
    [2, 4, 5, 7]
    [3, 6]
    [1]
    “””

    def helper(lst, nxt, curr):
        if lst:
            for t in lst:
                nxt += branches(t)
                curr.append(root(t))
            helper(nxt, [], [])
            print(curr)

    helper([t], [], [])
```