Iterators

Announcements

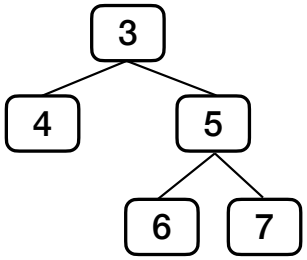
Discussion 5

Discussion 5: Make Path

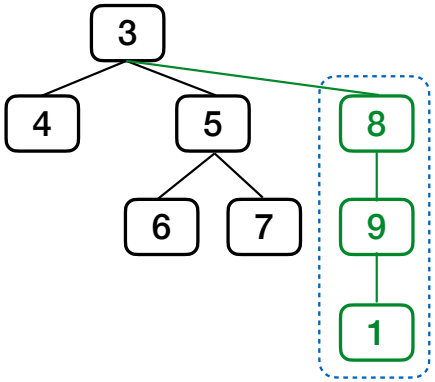
Implement `make_path`, which takes a tree `t` with unique labels and a list `p` that starts with the root label of `t`. It returns the tree `u` with the fewest nodes for which

1. `has_path(u, p)` returns `True`
2. `has_path(u, q)` returns `True` for all lists `q` for which `has_path(t, q)` returns `True`

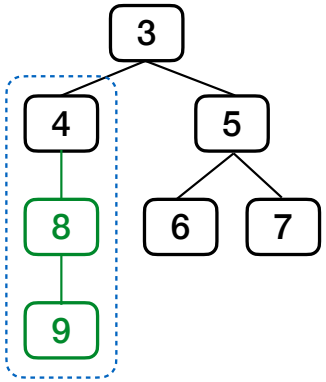
`t1`



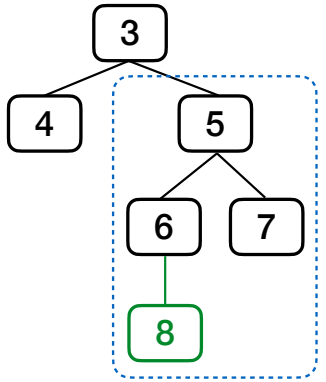
`make_path(t1, [3,8,9,1])`



`make_path(t1, [3,4,8,9])`



`make_path(t1, [3,5,6,8])`



Recursive idea: `make_path(b, p[1:])` is a branch of the tree returned by `make_path(t, p)`

Special case: if no branch starts with p[1], then a leaf labeled p[1] needs to be added

(Demo)

Tree Practice

Spring 2023 Midterm 2 Question 4(a)

Implement exclude, which takes a tree t and a value x. It returns a tree containing the root node of t as well as each non-root node of t with a label not equal to x. The parent of a node in the result is its nearest ancestor node that is not excluded.

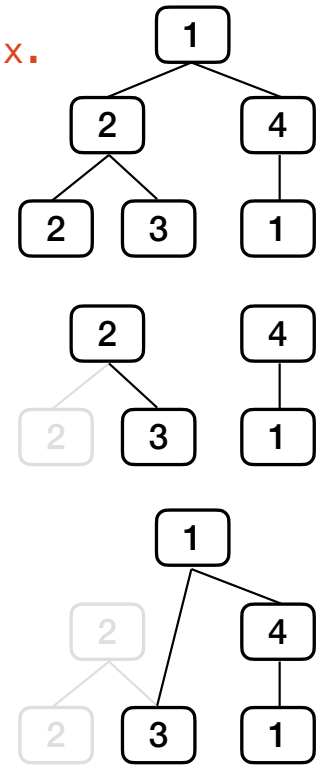
```
def exclude(t, x):
    """Return a tree with the non-root nodes of tree t labeled anything but x.

    >>> t = tree(1, [tree(2, [tree(2), tree(3)]), tree(4, [tree(1)])])
    >>> exclude(t, 2)
    [1, [3], [4, [1]]]
    >>> exclude(t, 1)  # The root node cannot be excluded
    [1, [2, [2], [3]], [4]]
    """
    filtered_branches = map(lambda y: _____________, branches(t))
    bs = []
    for b in filtered_branches:
        if _____________:
            _____________________
        else:
            bs.append(b)
    return tree(label(t), bs)
```

exclude(y, x)

label(b) == x

bs.extend(branches(b))



Tuples

(Demo)

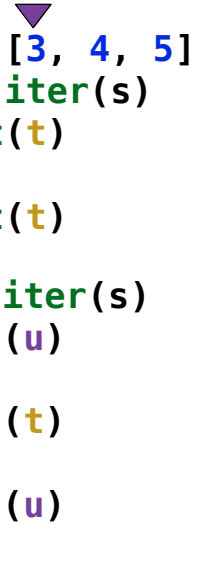
Iterators

Iterators

A container can provide an iterator that provides access to its elements in order

`iter(iterable)`: Return an iterator over the elements of an iterable value

`next(iterator)`: Return the next element in an iterator



```
>>> s = [3, 4, 5]
>>> t = iter(s)
>>> next(t)
3
>>> next(t)
4
>>> u = iter(s)
>>> next(u)
3
>>> next(t)
5
>>> next(u)
4
```

(Demo)

Discussion Question

What will be printed?

```
a = [1, 2, 3]
b = [a, 4]
c = iter(a)
d = c
print(next(c))
print(next(d))
print(b)
```

Map Function

(Demo)

Discussion Question

all(s) iterates through s until a false value is found (or the end is reached).

What's printed when evaluating:

```
x = all(map(print, range(-3, 3)))
```