Daily Coding Problem Blog

Daily Coding Problem #116

Problem

This problem was asked by Jane Street.

Generate a finite, but an arbitrarily large binary tree quickly in O(1).

That is, `generate()` should return a tree whose size is unbounded but finite.

Solution

Eager tree generation

If we ignore the O(1) generation constraint, we can create an unbounded tree by using randomness.

That is, we can generate the `left` and `right` sub-trees recursively `X`% of the time.

Since the question didn't have any constraint about the values the nodes can have, it's arbitrarily set to `0`.

```
import random

class Node:
    def __init__(self, val, left=None, right=None):
        self.val = val
        self.left = left
        self.right = right
```

```
def generate():
    root = Node(0)

    if random.random() < 0.5:
        root.left = generate()
    if random.random() < 0.5:
        root.right = generate()

    return root
```

Lazy tree generation

The trick here is that we can generate the tree lazily. Here we use Python's `property` keyword, which lets us define a property that of an object at look-up time.

When a `left` or `right` property is looked up, we check if that sub-tree has been evaulated. If not, we recursively create a new node half the time. If it has been, then we just return that node.

The object is O(1) to create since nothing happens when it's created.

```
class Node:
    def __init__(self, val, left=None, right=None):
        self.val = val
        self._left = left
        self._right = right

        self._is_left_evaluated = False
        self._is_right_evaluated = False


    @property
    def left(self):
        if not self._is_left_evaluated:
            if random.random() < 0.5:
                self._left = Node(0)

        self._is_left_evaluated = True
        return self._left
    @property
    def right(self):
        if not self._is_right_evaluated:
            if random.random() < 0.5:
                self._right = Node(0)

        self._is_right_evaluated = True
        return self._right

def generate():
    return Node(0)
```

© Daily Coding Problem 2019

Privacy Policy

Terms of Service

Press