Back

Python beats 95%, using distance to leaf

yaazi ★ 2 Last Edit: January 6, 2020 1:09 PM 104 VIEWS

1

Idea is to calculate the distance to leaf for each node and store the list of nodes at the same distance in a dictionary. Distance is the maximum distance from the left and right nodes. Then iterate over the depths in increasing order and append the node values to a list

```
from collections import defaultdict

class Solution(object):
    def findLeaves(self, root):
        dists = defaultdict(list)

        def dist_to_leaf(node):
            if not node:
                return -1
            maxdist = max(1 + dist_to_leaf(node.left), 1 + dist_to_leaf(node.right))
            dists[maxdist].append(node.val)
            return maxdist

        dist_to_leaf(root)
        res = []
        d = 0 # start with distance to leaf to 0, increase till not found in dict
        while d in dists:
            res.append(dists[d])
            d += 1

        return res
```

python dfs simple solution