



# Daily Coding Problem #107

## Problem

This problem was asked by Microsoft.

Print the nodes in a binary tree level-wise. For example, the following should print 1, 2, 3, 4, 5.

```
  1
 / \
2   3
 / \
4   5
```

## Solution

We can solve this problem by using a queue, initialized with the root, and continuously grabbing the first element and adding its left child and then its right child to the back of the queue, like so:

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

from queue import Queue

def print_level_order(root):
    queue = Queue()
    queue.put(root)
```

```
while not queue.empty():
    node = queue.get()
    if node.left:
        queue.put(node.left)
    if node.right:
        queue.put(node.right)
    print(node.val)

root = Node(1, Node(2), Node(3, Node(4), Node(5)))
print_level_order(root)
```

This takes  $O(N)$  time and space, since we have to look at the whole tree.

---

© Daily Coding Problem 2019

[Privacy Policy](#)

[Terms of Service](#)

[Press](#)