



Daily Coding Problem #83

Problem

This problem was asked by Google.

Invert a binary tree.

For example, given the following tree:

```
    a
   / \
  b   c
 / \ /
d  e f
```

should become:

```
    a
   / \
  c   b
 \  / \
 f e  d
```

Solution

Assuming we could invert the current node's left and right subtrees, all we'd need to do is then switch the left to now become right, and right to become left. The base case is when the node is None and we can just return None for that case. Then we know this works for the leaf node case since switching left and right subtrees doesn't do anything (since they're both None).

```
def invert(node):
    if not node:
        return node
```

```
left = invert(node.left)
right = invert(node.right)

node.left, node.right = right, left
return node
```

© Daily Coding Problem 2019

[Privacy Policy](#)

[Terms of Service](#)

[Press](#)