class Node:

def \_\_init\_\_(self, value):

self.data = value

self.left\_child = None

self.right\_child = None

def invert\_binary\_tree(node):

if node is None:

return

else:

# this judge is unnecessary

if (node.left\_child is None) and (node.right\_child is None):

return

else:

node\_temp = node.left\_child

node.left\_child = node.right\_child

invert\_binary\_tree(node.left\_child)

node.right\_child = node\_temp

invert\_binary\_tree(node.right\_child)

def inorder(root):

if root is not None:

inorder(root.left)

print(root.key,)

inorder(root.right)

root = Node(9)

root.left\_child = Node(6)

root.right\_child = Node(3)

root.left\_child.left\_child = Node(5)

root.left\_child.right\_child = Node(7)

root.left\_child.right\_child.left\_child = Node(15)

root.left\_child.right\_child.right\_child = Node(1)

root.right\_child.right\_child = Node(2)