BST to linkedlist

##Pretty simple, use a dict to store all nodes in pre-trans. Connect them one by one.

# Definition for a binary tree node.

# class TreeNode(object):

# def \_\_init\_\_(self, x):

# self.val = x

# self.left = None

# self.right = None

class Solution(object):

def convert(self, root):

"""

:type root: TreeNode

:rtype: TreeNode

"""

node = node2 = root

lst = []

def rec(root):

if root:

rec(root.left)

lst.append(root)

rec(root.right)

rec(root)

if len(lst)<2:

return root

if len(lst)==2:

root.left.right = root

return root

#print lst,len(lst)

for i in range(1,len(lst)-1):

lst[i].left = lst[i-1]

lst[i].right = lst[i+1]

lst[0].right,lst[-1].left = lst[1],lst[-2]

return lst[0]