Tree

Dec 28, 2018 12:06 AM, if — twice, just black it

Finish these two page, 23 probelms on 28. Fighting.

Pre-order

~~99~~ Recover Binary Search Tree (/prob ch-tree)

```
Input: [3,1,4,null,null,2]

  3
 / \
1   4
   /
  2

Output: [2,1,4,null,null,3]

  2
 / \
1   4
   /
  3
```

constant space one pass solution

def recoverTree(self, root):
Two possibilities:
1. Two 错位
2. One 错位, as the example, 1 3 2 4

6, 3, 4, 5, 2

dislocation

```
if cur.val < prev.val:
    if first == None:
        first = prev
    if first != None:
        second = cur
```

Dec 28, 2018 2:48 AM

105 Construct Binary Tree from Preorder and Inorder Traversal

The solution didn't use recursion,

106 Construct Binary Tree from Inorder and Postorder Traversal (/problems/construct-binary-tree-from-inorder-and-postorder-traversal)

✔ ~~111~~ Minimum Depth of Binary Tree

```
  3
 / \
9  20
  /  \
 15   7
```

return 2, BFS 遇到叶节点立即返回的做法

Dec 28, 2018 2:57 AM

```
if cur.left == None and cur.right == None:
    return height + 1
```

```
    1
   / \
  2   5
 / \   \
3   4   6
```

```
1
 \
  2
   \
    3
     \
      4
       \
        5
         \
          6
```

```
self.flatten(root.right)
self.flatten(root.left)
root.right = self.prev
root.left = None
self.prev = root
```

✔ ~~114~~ Flatten Binary Tree to Linked List

✔ ~~117~~ Populating Next Right Pointers in Each Node II (/ ng right-pointers-in-each-node-ii) Can skip

```
      1 -> NULL
     /  \
    2 -> 3 -> NULL
   / \     \
  4-> 5 -> 7 -> NULL
```

```
for i in range(size):
    cur = dq.popleft()
    if cur.left != None:
        dq.append(cur.left)
    if cur.right != None:
        dq.append(cur.right)

for i in range(len(dq) - 1):
    dq[i].next = dq[i+1]
```

✔ ~~124~~ Binary Tree Maximum Path Sum

```
Input: [1,2,3]

      1
     / \
    2   3

Output: 6
```

```
Input: [-10,9,20,null,null,15,7]

   -10
   / \
  9  20
    /  \
   15   7

Output: 42
```

```
Input: [1,2,3,null,5,null,4]
Output: [1, 3, 4]
Explanation:

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

~~199~~ ~~Binary Tree Right Side View~~ (/problems/binary-tree-right-side-view)

Can skip

```
Input:
     1
   /   \
  2     3
 / \   /
4   5 6

Output: 6
```

~~222~~ Count Complete Tree Nodes (/problems/count-complete-tree-nodes)

In a complete binary tree every level, except possibly the last, is completely filled, and all nodes in the last level are as far left as possible. It can have between 1 and 2h nodes inclusive at the last level h.

Utilize left depth and right depth, if they're same, return (1 << leftDepth) - 1

✔ ~~250~~ ~~Count Univalue Subtrees~~ ~~e-subtree~~

```
Input:  root = [5,1,5,5,5,null,5]

          5
         / \
        1   5
       / \   \
      5   5   5

Output: 4
```

~~255~~ Verify Preorder Sequence in Binary Search Tree

```
Input: [5,2,1,3,6]
Output: true
```

```
if len(preorder) == 0:
    return True
from collections import deque

dq = deque()
dq.append(preorder[0])
min_ = float('-inf')

for i in range(1, len(preorder)):
    if preorder[i] < min_:
        return False

    while len(dq) > 0 and dq[-1] < preorder[i]:
        min_ = dq[-1]
        dq.pop()
    dq.append(preorder[i])
return True
```

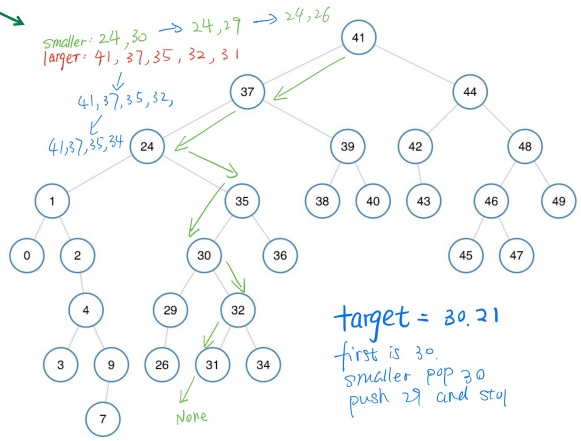
✔ 272 Closest Binary Search Tree Value II (/pr sest-binar value-ii)

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

target = 3.71428
Output: 4
```

✔ ~~298~~ Binary Tree Longest Consecutive Sequence (/problems/bina consecutive-sequence)

333 Largest BST Subtree (/proble)

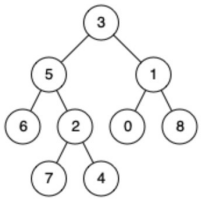
```
Input: [10,5,15,1,8,null,7]

   10
   / \
  5  15
 / \   \
1   8   7

Output: 3
```



~~156~~ Binary Tree Upside Down ary-tree-upside-do

```
Input:

   1
  /  \
 2    3
  \
   5

Output: ["1->2->5", "1->3"]
```

```
Input: [1,2,3,4,5]

    1
   / \
  2   3
 / \
4   5

Output: return the

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



✔ ~~257~~ Binary Tree Paths (/proble -paths)

Can skip

1. If root == none or root in (p,q) , then return root
2. If left and right both not none, then return root
3. If one of left or right is not none return that one
4. If they are both none, return none

✔ ~~236~~ Lowest Common Ancestor of a Binary Tree (/problems/lowest-common-ancestor-of-a-binary-tree)

```
Input: root = [5,3,6,2,4,null,null,1], k = 3
       5
      / \
     3   6
    / \
   2   4
  /
 1
Output: 3
```

✔ ~~297~~ Serialize and Deserialize Binary Tree (/problems/serialize-and-deserialize-binary-tree)

The way use count

Can skip

✔ ~~230~~ Kth Smallest Element in a BST

Basically preorder, can skip, two ways, count or just preorder traverse.

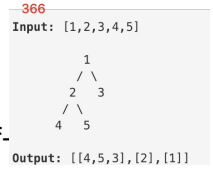
✔ ~~108~~ Convert Sorted Array to Binary Search Tree (/problems/convert-sorted-array-binary-tree)

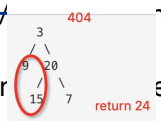
Can skip

```
Given the sorted array: [-10,-3,0,5,9],

One possible answer is: [0,-3,9,-10,nul

      0
     / \
   -3   9
   /   /
 -10  5
```

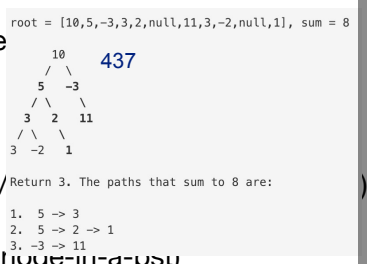
✔ ~~285~~ ~~Inorder Successor in BST~~ Similar idea with 272, can skip.

✔ ~~366~~ ~~Find Leaves of Binary Tree (/~~ ~~nd-leaves-of-~~



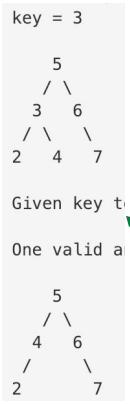


404 Sum of Left Leaves (/probler e



437

437 Path Sum III (/problems/path-sum-iii)



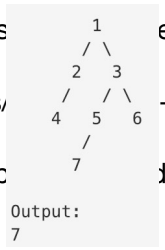
449 Serialize and Deserialize BST (/problems/)

Similar idea as 255

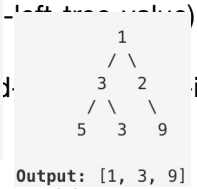
450 Delete Node in a BST (/problems/delete-node-in-a-bst)

501 Find Mode in Binary Search Tree (/problems/find-mode-in-binary-search-tree) find all the mode(s) (the most frequently occurred element) in the given BST.

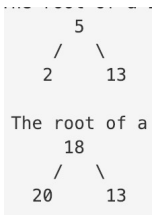
508 Most Frequent Subtree Sum (/problems ent-subtree-sum)



~~513~~ ~~Find Bottom Left Tree Value (/problems/ -left-tree-value)~~



~~515~~ ~~Find Largest Value in Each Tree Row~~ (/p -in-each-tree-row)



536 Construct Binary Tree from String (/problems/construct-binary-tree-from-string) 🔓

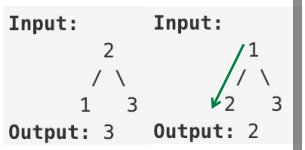
~~543~~ ~~Diameter of Binary Tree~~ (/problems/diameter-of-binary-tree)

The diameter of a binary tree is the length of the longest path between any two nodes in a tree.

538 Convert BST to Greater Tree (/problems/convert-bst-to-greater-tree)

every key of the original BST is changed to the original key plus sum of all keys greater than the original key in BST.

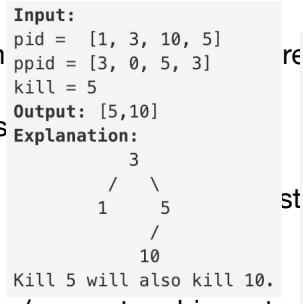
545 Boundary of Binary Tree (/problems/boundary-of-binary-tree) 🔓



549 Binary Tree Longest Consecutive Sequence II (/problems/binary-tree-longest-consecutive-sequence-ii) 🔓

the path can be in the child-Parent-child order,

563 Binary Tree Tilt (/problems/binary-tree-tilt)

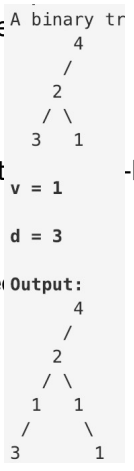


~~572~~ ~~Subtree of Another Tree~~ (/problem re

Same to same tree

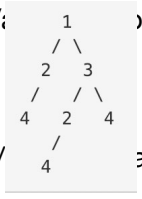
582 Kill Process (/problems/kill-proces

606 Construct String from Binary Tree st -binary-tree)



✔ ~~617~~ ~~Merge Two Binary Trees~~ (/problems/merge-two-binary-tre

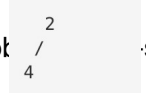
623 Add One Row to Tree (/problems/ ow-to-tree)



~~637~~ ~~Average of Levels in Binary Tree~~ (/ average-of-levels-in-binary-tree)

652 Find Duplicate Subtrees (/problem licate-subtrees)

following are two



~~653~~ ~~Two Sum IV - Input is a BST~~ (/prob -sum-iv-input-is-a-bst)

654 Maximum Binary Tree (/problems/maximum-binary-tree)

655 Print Binary Tree (/problems/print-binary-tree)

662 Maximum Width of Binary Tree (/problems/maximum-width-of-binary-tree)

663 Equal Tree Partition (/problems/equal-tree-partition) 🔓

666 Path Sum IV (/problems/path-sum-iv) 🔓

✔ 669 Trim a Binary Search Tree (/problems/trim-a-binary-search-tree)

671 Second Minimum Node In a Binary Tree (/problems/second-minimum-node-in-a-binary-tree)

684 Redundant Connection (/problems/redundant-connection)

685 Redundant Connection II (/problems/redundant-connection-ii)

687 Longest Univalue Path (/problems/longest-univalue-path)

742 Closest Leaf in a Binary Tree (/problems/closest-leaf-in-a-binary-tree) 🔓

814 Binary Tree Pruning (/problems/binary-tree-pruning)

834 Sum of Distances in Tree (/problems/sum-of-distances-in-tree)

863 All Nodes Distance K in Binary Tree (/problems/all-nodes-distance-k-in-binary-tree)

865 Smallest Subtree with all the Deepest Nodes (/problems/smallest-subtree-with-all-the-deepest-nodes)

426 Convert Binary Search Tree to Sorted Doubly Linked List (/problems/convert-binary-search-tree-to-sorted-doubly-linked-list) 🔓

701 Insert into a Binary Search Tree (/problems/insert-into-a-binary-search-tree)

700 Search in a Binary Search Tree (/problems/search-in-a-binary-search-tree)

590 N-ary Tree Postorder Traversal (/problems/n-ary-tree-postorder-traversal)

589 N-ary Tree Preorder Traversal (/problems/n-ary-tree-preorder-traversal)

429 N-ary Tree Level Order Traversal (/problems/n-ary-tree-level-order-traversal)

559 Maximum Depth of N-ary Tree (/problems/maximum-depth-of-n-ary-tree)

431 Encode N-ary Tree to Binary Tree (/problems/encode-n-ary-tree-to-binary-tree) 🔓

428 Serialize and Deserialize N-ary Tree (/problems/serialize-and-deserialize-n-ary-tree) [?]

872 Leaf-Similar Trees (/problems/leaf-similar-trees)

889 Construct Binary Tree from Preorder and Postorder Traversal (/problems/construct-binary-tree-from-preorder-and-postorder-traversal)

894 All Possible Full Binary Trees (/problems/all-possible-full-binary-trees)

897 Increasing Order Search Tree (/problems/increasing-order-search-tree)

919 Complete Binary Tree Inserter (/problems/complete-binary-tree-inserter)

951 Flip Equivalent Binary Trees (/problems/flip-equivalent-binary-trees)

958 Check Completeness of a Binary Tree (/problems/check-completeness-of-a-binary-tree)