

501. Find Mode in Binary Search Tree



Description (?tab=Description)



Submission (?tab=Submission)



Solutions (?tab=Solutions)



Add to List (?tab=List)

Total Accepted: **4437** Total Submissions: **11215** Difficulty: **Easy** Contributors: **Coder_1215** (/coder_1215/)

Notes

Given a binary search tree (BST) with duplicates, find all the mode(s) ([https://en.wikipedia.org/wiki/Mode_\(statistics\)](https://en.wikipedia.org/wiki/Mode_(statistics))) (the most frequently occurred element) in the given BST.

Assume a BST is defined as follows:

- The left subtree of a node contains only nodes with keys **less than or equal to** the node's key.
- The right subtree of a node contains only nodes with keys **greater than or equal to** the node's key.
- Both the left and right subtrees must also be binary search trees.

For example:

Given BST [1,null,2,2] ,

```

  1
   \
    2
   /
  2

```

return [2] .

Note: If a tree has more than one mode, you can return them in any order.

Follow up: Could you do that without using any extra space? (Assume that the implicit stack space incurred due to recursion does not count).

Subscribe (/subscribe/) to see which companies asked this question.

Show Tags

Show Similar Problems

Have you met this question in a real interview?

Discuss (<https://discuss.leetcode.com/category/650>)

Pick One (/problems/random-one-question/)

Editorial Solution

Python



```

1 # Definition for a binary tree node.
2 # class TreeNode(object):
3 #     def __init__(self, x):
4 #         self.val = x
5 #         self.left = None
6 #         self.right = None
7
8 class Solution(object):
9     def findMode(self, root):
10         """
11         :type root: TreeNode
12         :rtype: List[int]
13         """
14         if root==None:
15             return []
16         d={}
17
18         def preorder(root,traversal):
19             if root:
20                 traversal.append(root.val)
21                 preorder(root.left,traversal)
22                 preorder(root.right,traversal)
23             return traversal
24
25         nums=preorder(root,[])
26         for num in nums:
27             if num in d:
28                 d[num]+=1
29             else:
30

```

Send Feedback (mailto:admin@leetcode.com?subject=Feedback)

```
32         d[num]=1
33
34         maxi=max(d.values())
35
36         modes=[]
37         for key,value in d.items():
38             if value==maxi:
39                 modes.append(key)
40
41         return modes
```


Custom Testcase Contribute Testcase 

Shortcut: Ctrl + enter

Run Code

Submit Solution

Notes

Submission Result: Accepted (/submissions/detail/92584598/) More Details  (/submissions/detail/92584598/)Next challenges: [\(M\) Validate Binary Search Tree \(/problems/validate-binary-search-tree/\)](/problems/validate-binary-search-tree/)

Share your acceptance!

 9[Frequently Asked Questions \(/faq/\)](/faq/) | [Terms of Service \(/tos/\)](/tos/)[Privacy](#)

Copyright © 2017 LeetCode

 Send Feedback (mailto:admin@leetcode.com?subject=Feedback)