**6** 💟 🗓

## 510. Inorder Successor in a BST II June 6, 2019 | 6.4K views

Average Rating: 4.92 (13 votes)

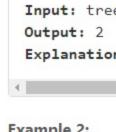
If that node has no in-order successor, return null.

The successor of a node is the node with the smallest key greater than node.val.

You will have direct access to the node but not to the root of the tree. Each node will have a reference to its parent node. Below is the definition for Node:

class Node { public int val; public Node left;

```
public Node right;
      public Node parent;
Follow up:
```



```
Input: tree = [5,3,6,2,4,null,null,1], node = 6
 Output: null
Example 3:
                             15
```



# Successor and Predecessor

current one.

before the current one.

Solution

# 34

25

successor

successor (= null successor) in such a situation.

inorder traversal = [1, 2, 7, 11, 12, 13, 25, 33, 34, 36, 40] 34

Successor of the node with a right child:

one step right and then left till you can

Successor of the node with no right child.

Go up till the node that is the left child of its parent, then return the parent.

node

inorder traversal = [1, 2, 7, 11, 12, 13, 25, 33, 34, 36, 40] null successor Successor of the node with no right child. Go up till the node that is the left child of its parent, then return the parent. inorder traversal = [1, 2, 7, 11, 12, 13, 25, 33, 34, 36, 40] node

Algorithm

C++

9 10

11 12

13

- Space complexity :  $\mathcal{O}(1)$ , since no additional space is allocated during the calculation. Analysis written by @liaison and @andvary Rate this article: \* \* \* \* \*
- Preview

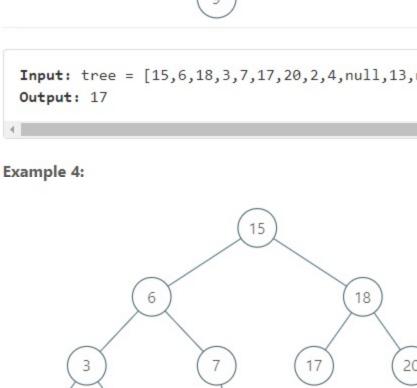
```
The first time I wrote almost exactly the same as the official solution:
 class Solution:
     def inorderSuccessor(self, node: 'Node') -> 'Node':
         if node right:
                                           Read More
2 A V C Share  Reply
theseungjin 🛊 94 🗿 January 20, 2020 9:47 AM
Can anyone explain clearly why "Go up til the node that is the left child of its parent" logic works? I can
rationalize this to myself knowing the fact but can't seem to logically explain.
1 A V C Share Share
```

Where is the "node in it"? Only the tree is given. Why is the method signature not the same as 285.

SHOW 4 REPLIES obaida ★0 ② 2 days ago

Inorder Successor in BST?

**Input:** tree = [2,1,3], node = 1 Explanation: 1's in-order successor node is 2. Note that both the node and the return Example 2:



All Nodes will have unique values.

Successor = "after node", i.e. the next node in the inorder traversal, or the smallest node after the

Predecessor = "before node", i.e. the previous node in the inorder traversal, or the largest node

inorder traversal = [1, 2, 7, 11, 12, 13, 25, 33, 34, 36, 40]

node

predecessor

## Approach 1: Iteration Intuition

Implementation Python Java 1 class Solution: def inorderSuccessor(self, node: 'Node') -> 'Node': # the successor is somewhere lower in the right subtree if node.right: node = node.right while node.left: node = node.left return node # the successor is somewhere upper in the tree while node.parent and node == node.parent.right: node = node.parent return node.parent

1. If the node has a right child, and hence its successor is somewhere lower in the tree. Go to the right

2. Node has no right child, and hence its successor is somewhere upper in the tree. Go up till the node

**Сору** 

Next **1** 

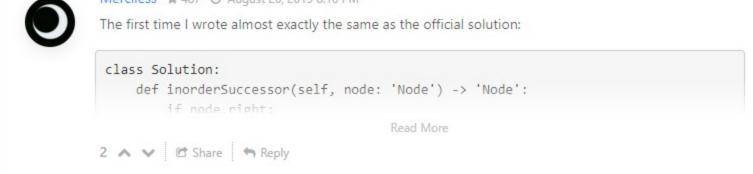
Sort By ▼

Post

once and then as many times to the left as you could. Return the node you end up with.

Comments: 4

O Previous



In Python soultion, line number 11 should be

Given a node in a binary search tree, find the in-order successor of that node in the BST.

Could you solve it without looking up any of the node's values? Example 1:

# Explanation: There is no in-order successor of the current node, so the answer is null

# 

• -10^5 <= Node.val <= 10^5 • 1 <= Number of Nodes <= 10^4

### There are two possible situations here: Node has a right child, and hence its successor is somewhere lower in the tree. To find the successor, go to the right once and then as many times to the left as you could.

successor . Node has no right child, then its successor is somewhere upper in the tree. To find the successor, go up

till the node that is left child of its parent. The answer is the parent. Beware that there could be no

successor

### **Complexity Analysis** ullet Time complexity : $\mathcal{O}(H)$ , where H is the height of the tree. That means $\mathcal{O}(\log N)$ in the average case, and $\mathcal{O}(N)$ in the worst case, where N is the number of nodes in the tree.

that is left child of its parent. The answer is the parent.

Type comment here... (Markdown is supported)

SHOW 2 REPLIES 1961 \* 123 ② August 30, 2019 4:04 AM "Given a binary search tree and a node in it"

while node.parent and node==node.parent.left: 0 A V C Share Share