

## Problem Domain

Write a function that takes a tree as input. Determine whether or not each value in the tree is divisible by 3, 5 or both 3 and 5.

Input:

5  
3  
0

Output:

Fizz  
Buzz  
Buzz  
Fizz

Big O(  
Time O(n)  
Space O(n)

Edge Case  
empty tree  
returns an empty tree (None)

## Algorithm

Given a binary tree as input and a node with value None create an empty tree named fizzbuzztree.

Set current\_node to input binary tree root node

Determine if current\_node value is divisible by 3, 5 or both. Assign appropriate value to current\_node value.

If current\_node left child is valid, invoke function fizzbuzztree.

Set fizzbuzztree to updated binary tree  
return fizzbuzztree.

Code

```
from tree import BinarySearchTree
def FizzBuzzTree(current_node, bst):
    fizz_buzz_tree = BinaryTree()
    if current_node is None:
        current_node = bst.binarytree.root
    current_node = set_fizz_buzz(current_node, bst)
    if current_node.left_child != None:
        fizz_buzztree(current_node.left_child, bst)
    if current_node.right_child != None:
        fizz_buzztree(current_node.right_child, bst)
    fizz_buzz_tree = bst
    return fizz_buzz_tree
```

Code

```
def set_fizz_buzz(node_value):
    if node_value % 5 == 0 and node_value % 3 == 0:
        return 'FizzBuzz'
    elif node_value % 3 == 0:
        return 'Fizz'
    elif node_value % 5 == 0:
        return 'Buzz'
    else:
        return node_value
```