# Challenge 4: Find the Height of a BST

Given the root to a Binary Search Tree, write a function to find the height of the tree. A solution is placed in the "solution" section for your help, but we would suggest you to solve it on your own first.

**We'll cover the following**



## Problem Statement #

Implement a function `findHeight(root)` which returns the height of a given binary search tree. An illustration is also provided for your understanding.

- *Height of a Node* — the number of edges between a node and its deepest descendent
- *Height of a Tree* — Height of its root node

Also, keep in mind that the height of an empty tree and leaf nodes is zero.

### Output #

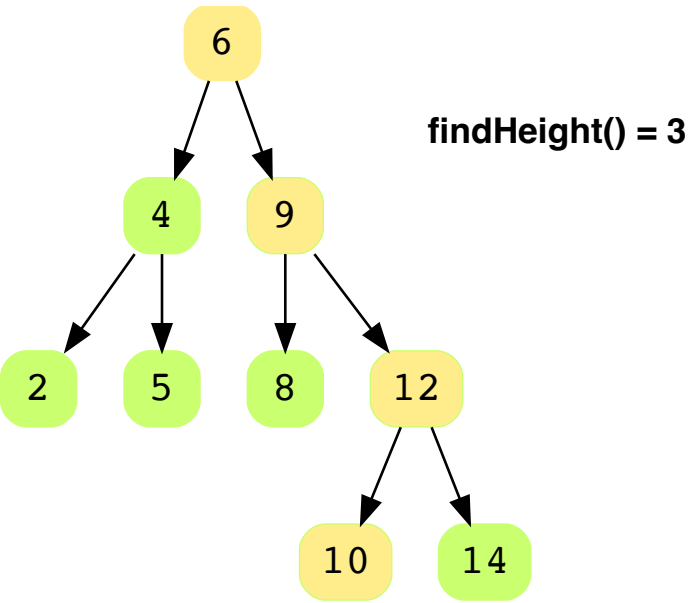
Returns the maximum depth or height of a binary tree

## Sample Input #

```
bst = {
        6 -> 4,9
    4 -> 2,5
    9 -> 8,12
    12 -> 10,14
}
where parent -> leftChild,rightChild
```

## Sample Output #

```
3
```




# Coding Exercise #

Take a close look and design a step-by-step algorithm first before jumping onto the implementation. This problem is designed for your practice, so try to solve it on your own first. If you get stuck, you can always refer to the solution provided in the solution section. Good Luck!

main.py

BinarySearchTree.py

Node.py

```
from Node import Node
from BinarySearchTree import BinarySearchTree


def findHeight(root, current_height = -1, max_height = -1):
    # Write your code here
    if root is None:
        return current_height
    max_height = max(current_height, max_height)
    max_height = findHeight(root.leftChild, current_height + 1, max_height)
    max_height = findHeight(root.rightChild, current_height + 1, max_height)
    return max_height
```



Back

Solution Review: Find Ancestors of a ...

Next

Solution Review: Find the Height of a ...

Completed

Report an Issue