



Challenge 3: Find Ancestors of a given node in a BST

If you are given the root to a Binary Search Tree and a node value "k", can you write a code to find the ancestor of that node? 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
 - Output
 - Sample Input
 - Sample Output
- Coding Exercise

Problem Statement

Implement the `findAncestors(root, k)` function which will find the ancestors of the node whose value is "k". Here `root` is the root node of a binary search tree and `k` is an integer value of node whose ancestors you need to find. An illustration is also given. Your code is evaluated on the tree given in the example.

Output

Returns all the ancestors of `k` in the binary tree in a Python list.



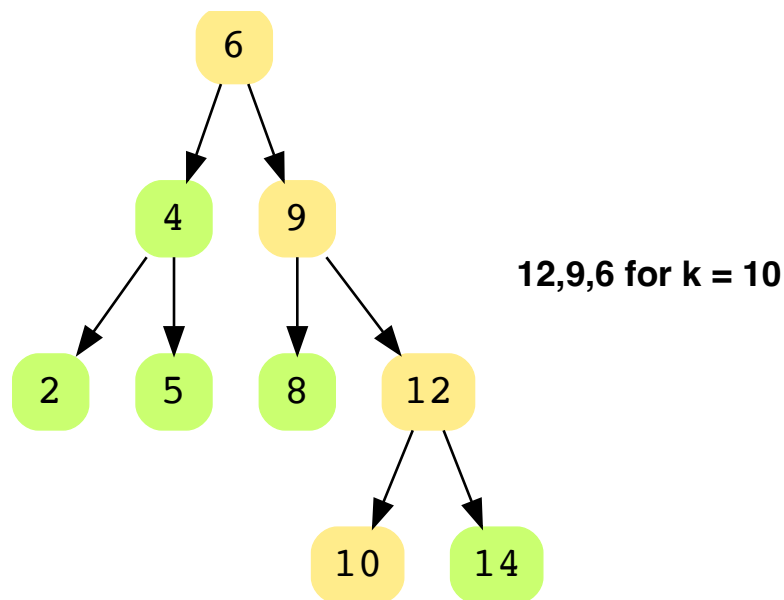
Sample Input



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

Sample Output

```
[12,9,6]
```



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
solution provided in the solution section. Good Luck!



main.py

BinarySearchTree.py

Node.py

```
from Node import Node
from BinarySearchTree import BinarySearchTree

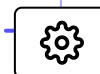
def findAncestors(root, k):
    # Write your code here
    if root is None:
        return None
    if root.val == k:
        return []

    if k < root.val:
        left = findAncestors(root.leftChild, k)
        if left is not None:
            left.append(root.val)
            return left
    else:
        right = findAncestors(root.rightChild, k)
        if right is not None:
            right.append(root.val)
            return right
    return None
```



Interviewing soon? We've partnered with Hired so that
companies apply to you instead of you applying to them. [See
how](#) ⓘ



[← Back](#)

Solution Review: Find kth maximum v...

Solution Review: Find Ancestors of a ...

 **Completed** [Report an Issue](#)