Brothers From Different Roots

Given two BSTs containing N**1** and N**2** distinct nodes respectively and given a value **x**. Your task is to complete the function **countPairs()**, that returns the count of all pairs from both the BSTs whose sum is equal to **x**.

Input : BST 1: 5
/ \
3 7
/ \ / \
2 4 6 8

```
BST 2:     10
          /   \
         6     15
       / \   /   \
      3   8  11   18
x = 16
```

Output : 3
The pairs are:
(5, 11), (6, 10) and **(8, 8)
**

```python
def countPairs(root1, root2, x):
    count =[0]
    helper(root1,x,count,root2)
    return count[0]

def helper(root,x,count,root2):
    if root is None:
        return
    helper(root.left,x,count,root2)
    target = x-root.data
    if search(root2,target):
        count[0] = count[0]+1
    helper(root.right,x,count,root2)



def search(root,x):
    if root is None:
        return False
    if root.data<x:
        return search(root.right,x)
    elif root.data>x:
        return search(root.left,x)
    else:
        return True
```