**Check if subtree :-**

Given two binary trees with head reference as **T**and **S** having at most **N** nodes. The task is to check if S is present as subtree in T.  
A subtree of a tree T1 is a tree T2 consisting of a node in T1 and all of its descendants in T1.

**Example:**

**S:**          10  
              /   \  
            4     6  
                 /  
             30

**T:**                  26  
                      /   \  
                    10   3  
                   /   \     \

               4       6     3  
                       /  
                    30

In above example S is subtree of T.

Please note that subtree has to be having same leaves non leaves.

   10  
  /  
20

For example, above tree is not subtree of

         10  
       /     \  
    20     50  
   /   \  
30   40

But a subtree of

         30  
       /     \  
    10     50  
   /    
20

**Example 1:**

**Input:**

T:      1          S:   3

     /   \          /

    2     3         4

  /  \    /

  N    N  4

**Output:** 1

**Example 2:**

**Input:**

T:      26       S:   26

      /   \      /  \

    10     N    10    N

  /    \        /  \

  20    30        20  30

 /  \          /  \

40   60        40  60

**Output:** 1

**Your Task:**  
Complete the **function isSubtree()**that takes two **nodes**as **parameter**and **returns**true or false.  
**Expected Time Complexity:**O(N).  
**Expected Auxiliary Space:**O(N).

**Constraints:**  
1 <= Number of nodes <= 104  
1 <= Value of nodes <= 104