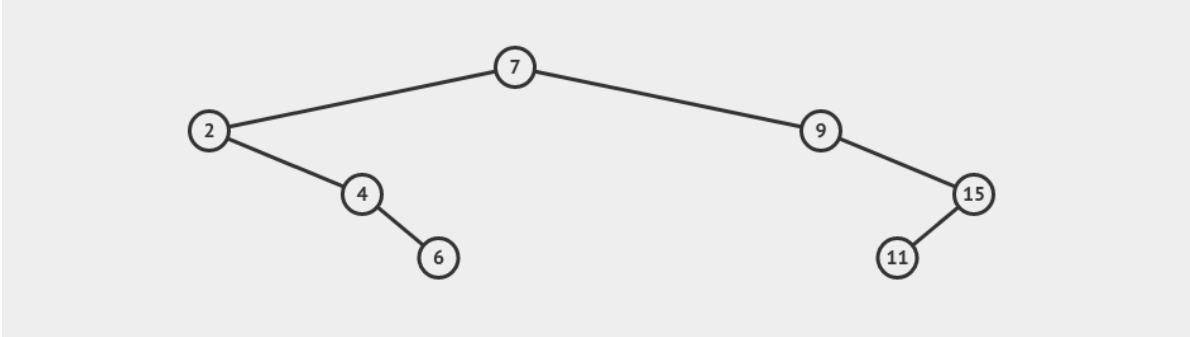
Members::

MIT2021072: Narayan Asati

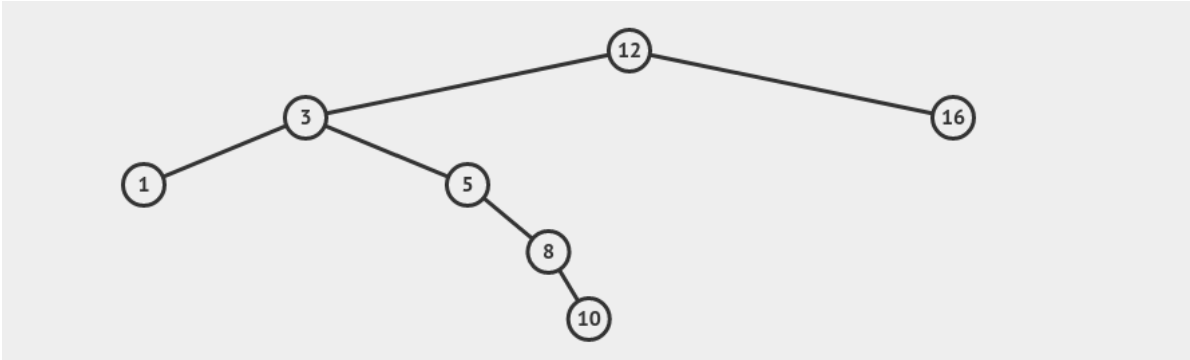
MIT2021117: Chandrakishor Singh

Q1. Represent the following sets by using BST: A={7,2,4,9,15,6,11}, B={12,3,16,1,5,8,10}. Merge both the sets into new set C (test if merge operation is possible). Write the procedure to display C. Since, we are using BST for representing the set, the internal BST representation of the sets A and B will be as follows.

A = {7,2,4,9,15,6,11}



B = {12,3,16,1,5,8,10}



We will use the following node structure for representing the node.

```
Node:
    data: The integer value stored in the node
    left: The pointer to left subtree
    right: The pointer to right subtree
```

We will use following structure for representing the BST.

```
BST:
    head: A pointer to the head of the BST. It is a structure of Node type.
```

Below is the pseudo code for the insert operation for BST.

```
# root: head of BST
# data: data of the new node to be inserted
INSERT(root, data):
    IF root == NULL:
        RETURN Node(data)

    IF (root.data > data)
        root.right = INSERT(root.right, data)
    IF (root.data < data)
        root.left = INSERT(root.left, data)
```

Below is the pseudo code for the *is_member(x)* function for BST.

```
# root: head of BST
# data: data of the node to be searched
IS_MEMBER(root, data):
    IF root == NULL:
        RETURN False

    IF (root.data > data)
        RETURN IS_MEMBER(root.right, data)
    IF (root.data < data)
        RETURN IS_MEMBER(root.left, data)
    RETURN TRUE
```

To merge set A and B, we need to first ensure that A and B are disjoint. Only then merge operation will be possible. Hence, we will iterate over set A and will check whether the current element of set A is present in the set B or not(using *IS_MEMBER()* function). Then, we will iterate over both the sets A and B one-by-one and insert the current element into set C. Its pseudo code is as follows.

```
# A: Set A
# B: Set B
# rootA: Root of the BST of set A
# rootB: Root of the BST of set B

FOR node IN A:
    IF IS_MEMBER(rootB, node.data)
        PRINT "Sets are not disjoint. Not possible to merge them."
        EXIT

rootC = Create a new BST

FOR node IN A:
    INSERT(rootC, node.data)

FOR node IN B:
```

```
    INSERT(rootC, node.data)
```

To display the contents of set C, we will just iterate over C using following inorder traversal pseudo code.

```
# rootC: Root of the BST of set C

INORDER(rootC):
    IF rootC IS NULL:
        RETURN

    INORDER(rootC.left)
    PRINT rootC.data
    INORDER(rootC.right)
```