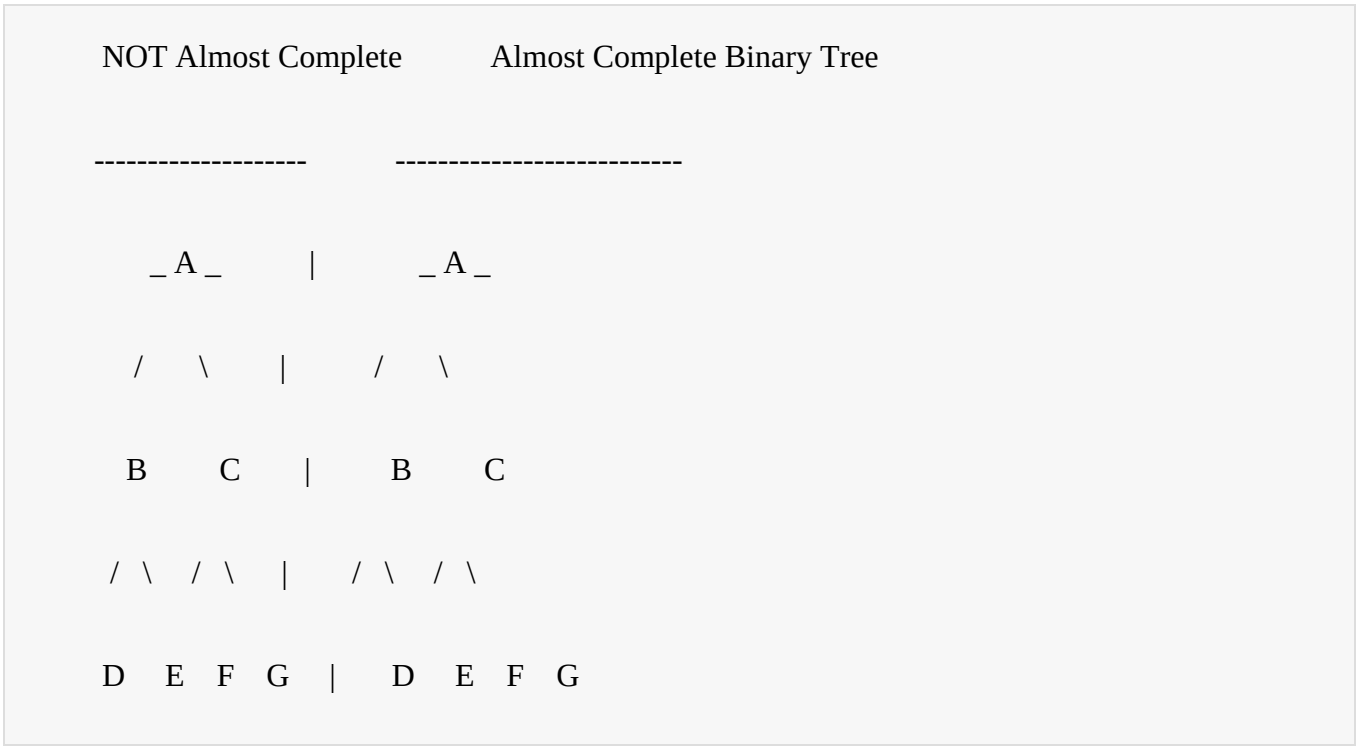
CHECK IF A TREE IS ALMOST COMPLETE BINARY TREE

Given a pointer to the root node of the tree, write code to find if it is an Almost Complete Binary Tree or not?

A Binary Tree of depth d is Almost Complete iff:

1. The tree is Complete Binary Tree (All nodes) till level (d-1).
2. At level d, (i.e the last level), if a Node is present, then all the Nodes to the left of that node should also be present.

For example, the left tree below is NOT an Almost Complete Binary Tree but the right tree is an Almost Complete Binary Tree



```
 /\    /\     |   /\

H  I   J  K    |  H  I
```

The left tree is not almost complete because in the last level, the nodes left to Node-J are missing (children of Node-E).

Solution:

The problem may look complex but the solution is simple. For this we will use Level-wise traversal of the Tree.

Traverse the tree in level-wise order and check for below two conditions:

– When first leaf node is found, then all the nodes following it should also be leaf nodes.

– If a Node has right child, then it should also have a Left child.

The 2nd condition is required for cases like below one

```
   A

   \

    B
```

Algorithm:

Traverse Nodes of the tree in level-wise order. For each node do the following:

1. If Node is leaf Node

   leafNodeFound = true

2. Else IF leafNodeFound AND Node is NOT leaf

   return FALSE;

3. ELSE IF Node has Right child but not left child

   return FALSE;

return TRUE;

Code:

```
1       bool checkAlmostComplete(root r)
2       {
3           // Enqueue the root in the Queue.
4           Q.enqueue(r);
5
6           bool leafFound = false;
7           while(!Q.empty())
```

```
8         {
9             // Print the top element in the Queue and insert its children
10            Node *temp = Q.dequeue();
11
12            if( (leafFound == true) && (temp->left != NULL || temp->right == NULL) )
13                return true;
14
15            if(temp->left == NULL && temp->right == NULL)
16                leafFound = true;
17
18            if(temp->left == NULL && temp->right != NULL)
19                return false;
20
21            if(temp->left)
22                Q.enqueue(temp->left);
23
24            if(temp->right)
25                Q.enqueue(temp-> right);
26        }
27        return true;
28    }
```

Source: http://www.ritambhara.in/check-if-a-tree-is-almost-complete-binary-tree/