

Python Code :-

```
class Node:

    # A utility function to create a new node
    def __init__(self, key):
        self.data = key
        self.left = None
        self.right = None

# Function to print level order traversal of tree
def printLevelOrder(root):
    h = height(root)
    for i in range(1, h+1):
        printCurrentLevel(root, i)

# Print nodes at a current level
def printCurrentLevel(root, level):
    if root is None:
        return
    if level == 1:
        print(root.data, end=" ")
    elif level > 1:
        printCurrentLevel(root.left, level-1)
        printCurrentLevel(root.right, level-1)

""" Compute the height of a tree--the number of nodes
    along the longest path from the root node down to
    the farthest leaf node
    """

def height(node):
    if node is None:
        return 0
    else:
        # Compute the height of each subtree
        lheight = height(node.left)
        rheight = height(node.right)

        # Use the larger one
        if lheight > rheight:
            return lheight+1
```

```
        else:
            return rheight+1

# Driver program to test above function
root = Node(1)
root.left = Node(2)
root.right = Node(3)
root.left.left = Node(4)
root.left.right = Node(5)

print("BFS of binary tree is -")
printLevelOrder(root)
```

Output:-

BFS of binary tree is

1 2 3 4 5