# 102、二叉树的层序遍历（Binary Tree Level Order Traversal）

## 题目：

给你一个二叉树，请你返回其按 **层序遍历** 得到的节点值。 （即逐层地，从左到右访问所有节点）。

**示例：**  
二叉树：[3,9,20,null,null,15,7],

3  
 / \  
 9 20  
 / \  
 15 7

返回其层次遍历结果：

[  
 [3],  
 [9,20],  
 [15,7]  
]

## 解答：

|  |
| --- |
| /\*   1 count rootQueue curRoot rootList  / \ 1  2 3 1) 1 2 3 1 1  / \ \ 2) 2 3 4 5 2 1/2  4 5 7 4 5 7 3 1/2-3  3) 3 5 7 4 1/2-3/4  7 5 1/2-3/4-5  7 1/2-3/4-5-7  \*/public IList<IList<int>> LevelOrder(TreeNode root)  {  IList<IList<int>> res = new List<IList<int>>();  if (root == null)  {  return res;  }  //先进先出的集合  Queue<TreeNode> rootQueue = new Queue<TreeNode>();  //添加当前的root  rootQueue.Enqueue(root);  //在他没有的时候退出循环  while (rootQueue.Count() > 0)  {  //区分不同的leve，一层一层的，把每一层的长度记录下来  int lenrootQueue = rootQueue.Count();  IList<int> rootList = new List<int>();  for (int i = 0; i < lenrootQueue; i++)  {  //每层有几个pop几个  TreeNode curRoot = rootQueue.Dequeue();  //放在level的集合中  rootList.Add(curRoot.val);  //加下次要用的值放近q  if (curRoot.left != null)  {  rootQueue.Enqueue(curRoot.left);  }  if (curRoot.right != null)  {  rootQueue.Enqueue(curRoot.right);  }  }  res.Add(rootList);  }  return res;  } |