# 107、二叉树的层次遍历 II（Binary Tree Level Order Traversal II）

## 题目：

给定一个二叉树，返回其节点值自底向上的层次遍历。 （即按从叶子节点所在层到根节点所在的层，逐层从左向右遍历）

例如：  
给定二叉树 [3,9,20,null,null,15,7],

3   
 / \   
 9 20   
 / \   
 15 7

返回其自底向上的层次遍历为：

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

## 解答：

|  |
| --- |
| 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);  }  }  //和102一样只是往前面插入即可  res.Insert(0, rootList);  }  return res;  } |