

思路：

首先，BFS,BFS对应的是层序遍历，最短路径

利用ARRAYDEQUE构建一个queue,removeFirst得到开头，AddLast加入结尾，

但是如果是BFS的话，我们输出的是3,9,20,15,7这样的1D array， 我们希望一层一层的遍历，

因此我们要把每一层作区分，

传统的BFS是Queue里有node就不停，这点我们不改

但是我们一次把queue内所有Node都处理完，这样就是一层，

class Solution {

    public List<List<Integer>> levelOrder(TreeNode root) {

        List<List<Integer>> result=new ArrayList<List<Integer>>();

        Deque<TreeNode> myQueue=new ArrayDeque<TreeNode>();

        if(root!=null)

            myQueue.addLast(root);

        while(!myQueue.isEmpty()){

            List<Integer> level=new ArrayList<Integer>();

            int n=myQueue.size(); //记录实际上这一层有几个node

            for(int i=0;i<n;i++){

                TreeNode temp=myQueue.removeFirst(); //处理这些node

                level.add(temp.val);

                if(temp.left!=null)

                    myQueue.addLast(temp.left);

                if(temp.right!=null)

                    myQueue.addLast(temp.right);

            }

            result.add(level);

        }

        return result;

    }

}