

155. 二叉树的最小深度

二叉树的最小深度为根节点到最近叶子节点的距离。叶子节点左右孩子都为空

样例:给出一棵如下的二叉树:

```
1
 / \
2  3
 / \
4  5
```

这个二叉树的最小深度为 2。 思路如下：

```
1 空节点 return 0
2 左右孩子都为空 则只有一个节点, return 1;
3 如果左孩子为空, 则在该节点的右孩子中找, 并加上该节点。
4 如果右孩子为空, 则在该节点的左孩子中找, 并加上该节点。
5 如果左右都不为空, 则左右都找, 并加上该节点。
```

```
1 public int minDepth(TreeNode root) {
2     // 1. recursion 递归
3     if( null == root ) return 0;
4     if( null == root.left && null == root.right ) return 1;
5     if( null == root.left ) return minDepth(root.right) + 1;
6     if( null == root.right ) return minDepth(root.left) + 1;
7     return Math.min( minDepth(root.left) + 1, minDepth(root.right) + 1 );
8 }
9 public int minDepth(TreeNode root) {
10    // 2.levelOrder 层序遍历, 第一个叶子节点的层数即为最小深度
11    if( null == root ) return 0;
12    int res = 0;
13    Queue<TreeNode> queue = new LinkedList<TreeNode>();
14    queue.offer(root);
15    while( !queue.isEmpty() ){
16        int levelsize = queue.size();
17        res++; //每一层在开始的时候访问一次
18        for( int i = 0; i < levelsize; i++ ){
19            TreeNode node = queue.poll();
20            if( null == node.left && null == node.right )
21                return res;
22            if( null != node.left ) queue.offer(node.left);
23            if( null != node.right ) queue.offer(node.right);
24        }
25    }
26    return res;
27 }
```

