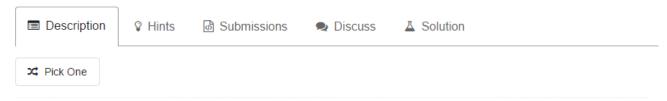
111. Minimum Depth of Binary Tree



Given a binary tree, find its minimum depth.

The minimum depth is the number of nodes along the shortest path from the root node down to the nearest leaf node.

Note: A leaf is a node with no children.

Example:

}

Given binary tree [3,9,20,null,null,15,7],

```
3
/\
9 20
/\
15 7
```

return its minimum depth = 2.

```
public class L111 {
    public class TreeNode {
       int val;
       TreeNode left;
       TreeNode right;
       public TreeNode(int x) {
           this.val = x;
    }
     public int minDepth(TreeNode root) {
         if (root == null) return 0;
           if (root.left == null && root.right == null) return 1;
            * 记住二叉树的深度是根节点到叶子节点的距离。重点,所以当left为null时,需要走右边去找。
            * 因为上面已经排除了root为叶子节点的可能 (left=null && right=null就是叶子节点)
           if (root.left == null) return minDepth(root.right) + 1;
           else if (root.right == null) return minDepth(root.left) + 1;
           else return 1 + Math.min(minDepth(root.left), minDepth(root.right));
     }
```