

111. Minimum Depth of Binary Tree

由 Xmind 截图

111. 二元樹的最小深度

難度 簡單 已 850 次 已 0 次 已 0 次 已 0 次

给定一个二叉树，找出其最小深度。

最小深度是从根节点到最近叶子节点的最短路径上的节点数量。

说明：叶子节点是指没有子节点的节点。

示例 1：

3

9

20

15

7

输入：root = [3,9,20,null,null,15,7]

输出：2

示例 2：

2

3

4

5

6

输入：root = [2,null,3,null,4,null,5,null,6]

输出：5

提示：

- 树中节点数的范围在 [0, 10^5] 内
- 10000 <= Node.val <= 10000

通过次数 448,627 提交次数 879,870

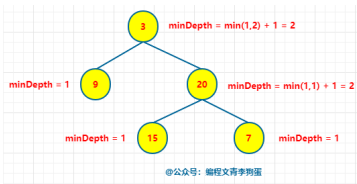
题解你在题解列表中看到过吗？ 讨论 帮助 举报 反馈

常規解法

自底向上

每次先遍歷左子樹，找出左子樹的最小深度，再遍歷右子樹，找出右子樹的最小深度，最終再根節點取左子樹和右子樹深度最小的那個，加上根節點的高度 1

min(leftMindepth, rightMindepth) + 1

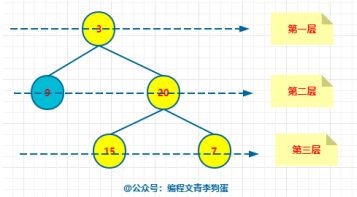


自左向右

就是從根節點開始，一層一層的遍歷二元樹。

這種一層一層的遍歷，其實用的就是之前講過的層次遍歷

當發現第 1 個左右節點都為空的節點（即葉子節點）的時候，它的深度就是二元樹的最小深度



遞迴法

1.找出重複的子問題。

第 2 種情況，只有左子樹或者只有右子樹的最小深度為“左子樹的最小深度 + 1” 或者“右子樹的最小深度 + 1”。

第 3 種情況，左子樹和右子樹都有的，最小深度為“min(左子樹的最小深度，右子樹的最小深度) + 1”

時間複雜度為 O(n)

2.確定終止條件。

二元樹為空，最小高度為 0
if root == None:
return 0

只有根節點，最小高度為 1
if root.left == None and root.right == None:
return 1

如果節點的左子樹不為空，右子樹為空
if root.left != None and root.right == None:
return leftMindepth + 1
如果節點的右子樹不為空，左子樹為空
if root.left == None and root.right != None:
return rightMindepth + 1

左右子樹都不為空
return min(leftMindepth, rightMindepth) + 1

非遞迴法（迭代）

```
class Solution:
    def minDepth(self, root: Optional[TreeNode]) -> int:
        # 1.確定終止條件re
        if root == None:
            return 0

        # 2.找出重複的子問題
        # 1.只有根節點，最小高度為 1
        if root.left == None and root.right == None:
            return 1

        # 左子樹最小值和右子樹最小值
        leftMinDepth = self.minDepth(root.left)
        rightMinDepth = self.minDepth(root.right)

        # 2.如果節點的左子樹不為空，右子樹為空
        if root.left != None and root.right == None:
            return leftMinDepth + 1
        # 3.如果節點的右子樹不為空，左子樹為空
        if root.right != None and root.left == None:
            return rightMinDepth + 1
        # 4.左右子樹都不為空
        return min(leftMinDepth, rightMinDepth) + 1
```