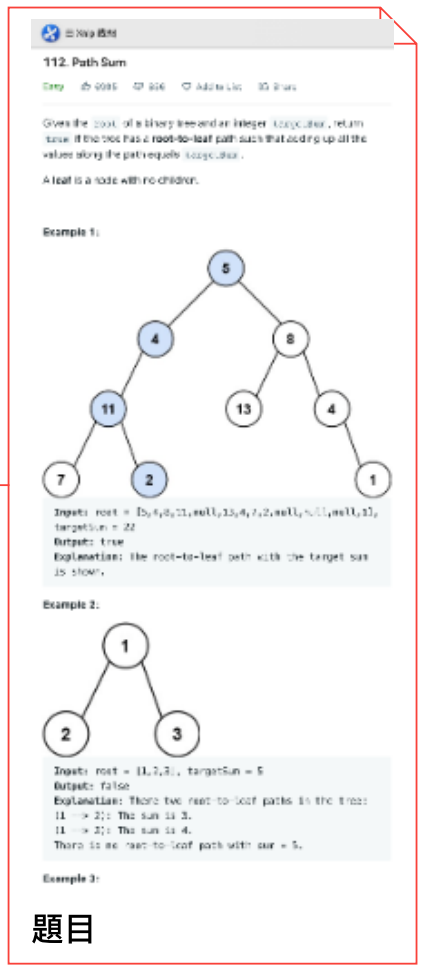
112. Path Sum

題目



112. Path Sum

Easy

Given the root of a binary tree and an integer targetSum, return true if the tree has a root-to-leaf path such that adding up all the values along the path equals targetSum.

A leaf is a node with no children.

Example 1:

Input: root = [5,4,8,11,null,13,4,7,2,null,null,null,1], targetSum = 22
Output: true
Explanation: The root-to-leaf path with the target sum is shown.

Example 2:

Input: root = [1,2,3], targetSum = 5
Output: false
Explanation: There two root-to-leaf paths in the tree:
(1 --> 2): The sum is 3.
(1 --> 3): The sum is 4.
There is no root-to-leaf path with sum = 5.

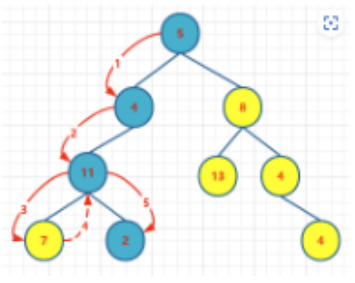
Example 3:

自頂向下用的其實就是前序遍歷

- 每次先判斷當前的節點
- 再遞迴左子樹
- 最後是右子樹

在本題路徑總和中，自頂向下的解法其實就是

- 判斷當前節點是否為葉子節點，如果是葉子節點，判斷當前葉子節點的值是否 = targetSum 減去之前路徑上節點值。
- 遞迴左子樹。
- 遞迴右子樹。



- 分支主題 4

遞迴法

遞迴二步曲

1. 找出重複的子問題。
   - 前序遍歷的順序是：根節點、左子樹、右子樹。
   - 本題同樣也是這個順序
     - 判斷節點
     - 遞迴左子樹：leftPath = self.hasPathSum(root.left, targetSum - root.val)
     - 遞迴右子樹：rightPath = self.hasPathSum(root.right, targetSum - root.val)
2. 確定終止條件。
   - 對於路徑來說，遍歷到葉子節點且葉子節點的值等於該路徑之前節點的值，證明已找到，返回 True。

```
# 如果當前節點為葉子節點，且葉子節點的值等於減去該路徑之前節點的值，返回 True
if root.left == None and root.right == None and root.val == targetSum:
    return True
```

題解

```
# Definition for a binary tree node.
# class TreeNode:
#     def __init__(self, val=0, left=None, right=None):
#         self.val = val
#         self.left = left
#         self.right = right
class Solution:
    def hasPathSum(self, root: Optional[TreeNode], targetSum: int) -> bool:
        # 如果樹為空，返回 False
        if root is None:
            return False
        # 如果當前節點為葉子節點，且葉子節點的值等於減去該路徑之前節點的值，返回 True
        if root.left is None and root.right is None and root.val == targetSum:
            return True
        # 遞迴左子樹
        left_path = self.hasPathSum(root.left, targetSum - root.val)
        # 遞迴右子樹
        right_path = self.hasPathSum(root.right, targetSum - root.val)

        return left_path or right_path
```