assignment8

Assignment #8: 🌲 (2/3)

Updated 2223 GMT+8 Oct 27, 2025

2025 fall, Complied by 杨知进 物理学院

> **说明：**
>
> 1. **解题与记录：**
>
>   对于每一个题目，请提供其解题思路（可选），并附上使用Python或C++编写的源代码（确保已在OpenJudge， Codeforces，LeetCode等平台上获得Accepted）。请将这些信息连同显示“Accepted”的截图一起填写到下方的作业模板中。（推荐使用Typora https://typoraio.cn 进行编辑，当然你也可以选择Word。）无论题目是否已通过，请标明每个题目大致花费的时间。
>
> 2. **提交安排：** 提交时，请首先上传PDF格式的文件，并将.md或.doc格式的文件作为附件上传至右侧的“作业评论”区。确保你的Canvas账户有一个清晰可见的本人头像，提交的文件为PDF格式，并且“作业评论”区包含上传的.md或.doc附件。
> 3. **延迟提交：** 如果你预计无法在截止日期前提交作业，请提前告知具体原因。这有助于我们了解情况并可能为你提供适当的延期或其他帮助。
>
> 请按照上述指导认真准备和提交作业，以保证顺利完成课程要求。

1. 题目

E108.将有序数组转换为二叉搜索树

https://leetcode.cn/problems/convert-sorted-array-to-binary-search-tree/

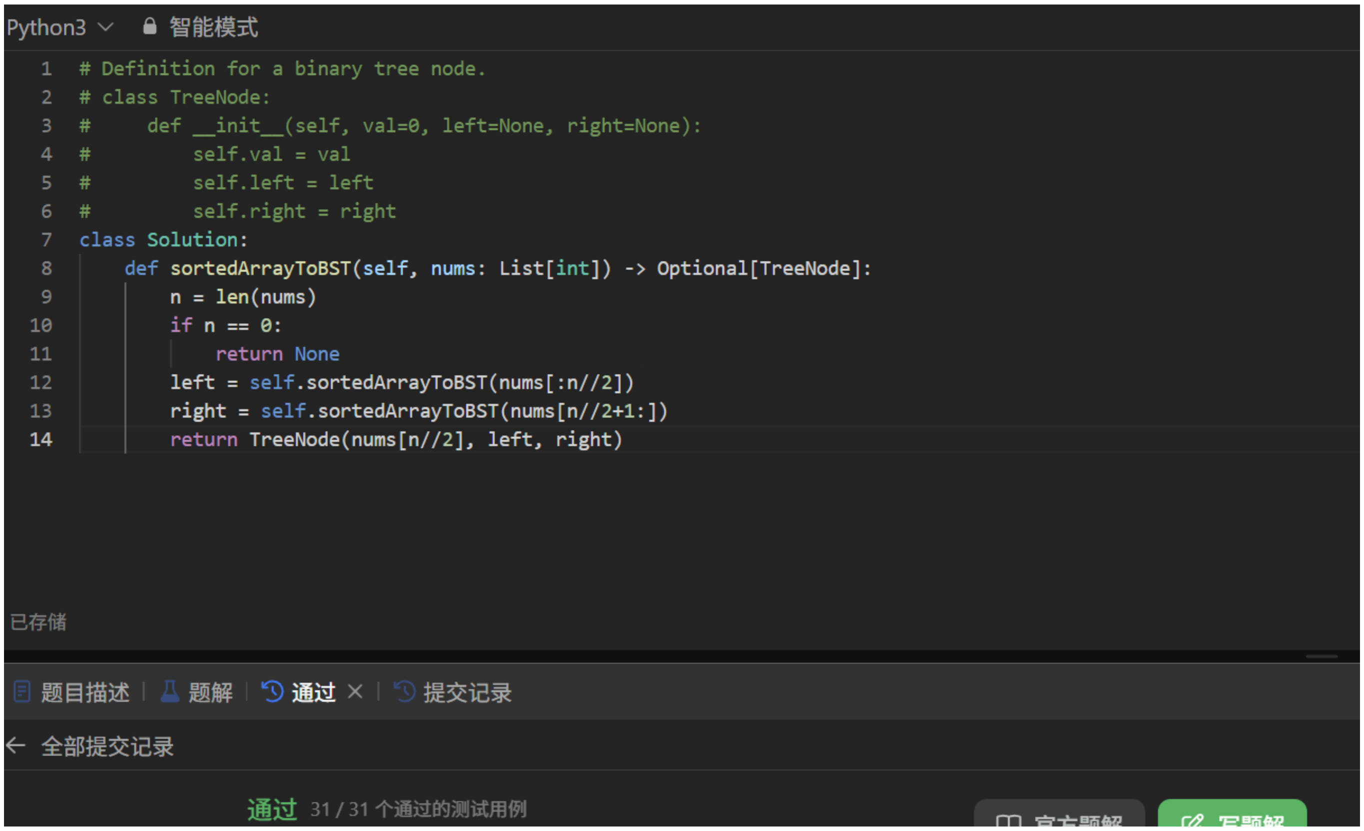
思路：

中间数作为根节点，递归

代码：

```python
class Solution:
    def sortedArrayToBST(self, nums: List[int]) -> Optional[TreeNode]:
        n = len(nums)
        if n == 0:
            return None
        left = self.sortedArrayToBST(nums[:n//2])
        right = self.sortedArrayToBST(nums[n//2+1:])
        return TreeNode(nums[n//2], left, right)
```

代码运行截图 （至少包含有"Accepted"）



```
# 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 sortedArrayToBST(self, nums: List[int]) -> Optional[TreeNode]:
        n = len(nums)
        if n == 0:
            return None
        left = self.sortedArrayToBST(nums[:n//2])
        right = self.sortedArrayToBST(nums[n//2+1:])
        return TreeNode(nums[n//2], left, right)
```

通过 31 / 31 个通过的测试用例

25fall Algo/week8/image.png

用时：18min左右

M07161: 森林的带度数层次序列存储

tree, http://cs101.openjudge.cn/practice/07161/

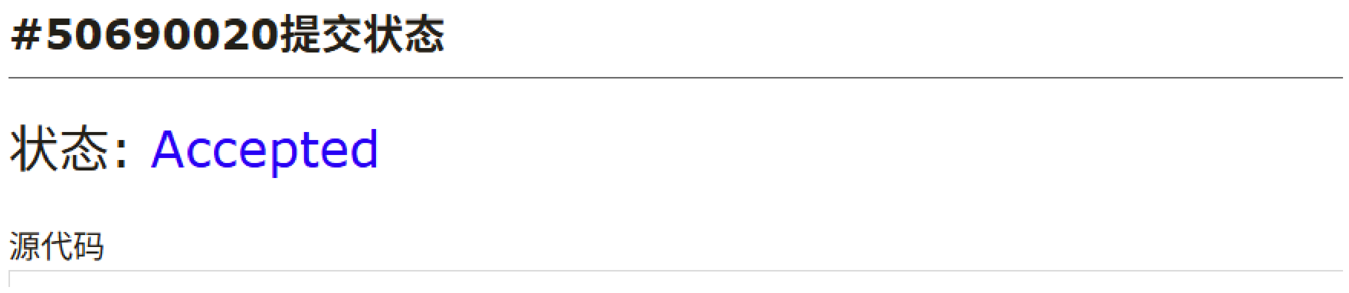
思路：

代码：

```
from collections import deque

class Node:
    def __init__(self, name, degree):
        self.name = name
        self.degree = degree
        self.children = []
```

```python
def build_tree(seq):
    tokens = seq.split()
    nodes = []
    for i in range(0, len(tokens), 2):
        name = tokens[i]
        degree = int(tokens[i+1])
        nodes.append((name, degree))

    if not nodes:
        return None

    root = Node(nodes[0][0], nodes[0][1])
    queue = deque()
    queue.append(root)
    index = 1
    while queue and index < len(nodes):
        current = queue.popleft()
        d = current.degree
        for _ in range(d):
            if index >= len(nodes):
                break
            child_name, child_degree = nodes[index]
            child_node = Node(child_name, child_degree)
            current.children.append(child_node)
            queue.append(child_node)
            index += 1
    return root

def postorder_traversal(root):
    result = []
    for child in root.children:
        result.extend(postorder_traversal(child))
    result.append(root.name)
    return result

def main():
    import sys
    data = sys.stdin.read().splitlines()
    n = int(data[0])
    trees = []
    for i in range(1, n+1):
        tree_seq = data[i]
        trees.append(tree_seq)

    forest_postorder = []
    for seq in trees:
        root = build_tree(seq)
        if root:
            forest_postorder.extend(postorder_traversal(root))

    print(" ".join(forest_postorder))

if __name__ == "__main__":
    main()
```

代码运行截图 ==（至少包含有"Accepted"）==



#50690020提交状态

状态: Accepted

源代码

image-3.png

用时：35miin左右

M27928: 遍历树

adjacency list, dfs, http://cs101.openjudge.cn/practice/27928/

思路：

代码：

```
```

代码运行截图 ==（至少包含有"Accepted"）==

M129.求根节点到叶节点数字之和

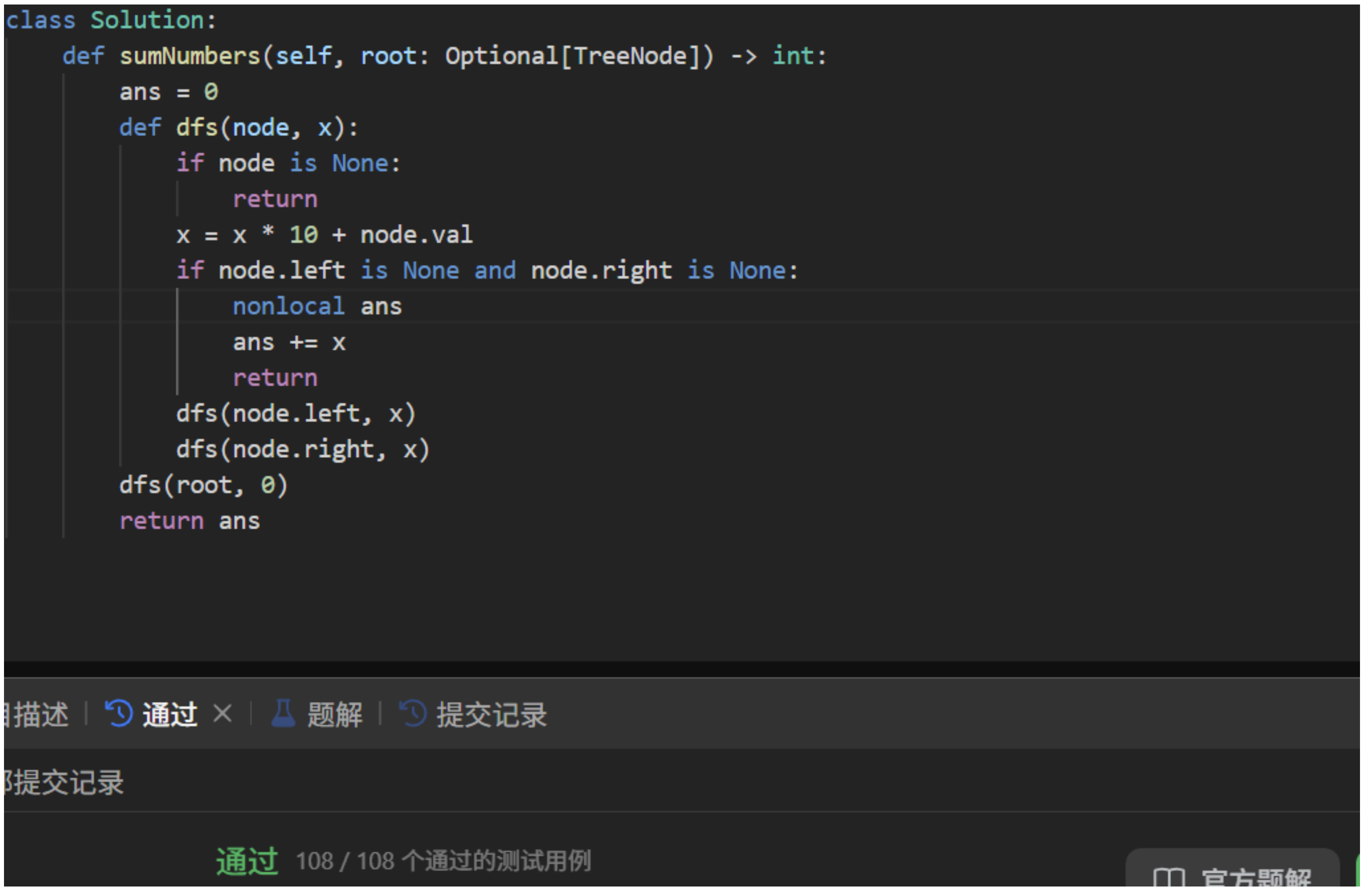
dfs, https://leetcode.cn/problems/sum-root-to-leaf-numbers/

思路：

代码

```python
class Solution:
    def sumNumbers(self, root: Optional[TreeNode]) -> int:
        ans = 0
        def dfs(node, x):
            if node is None:
                return
            x = x * 10 + node.val
            if node.left is None and node.right is None:
                nonlocal ans
                ans += x
                return
            dfs(node.left, x)
            dfs(node.right, x)
        dfs(root, 0)
        return ans
```

代码运行截图（至少包含有"Accepted"）



```
class Solution:
    def sumNumbers(self, root: Optional[TreeNode]) -> int:
        ans = 0
        def dfs(node, x):
            if node is None:
                return
            x = x * 10 + node.val
            if node.left is None and node.right is None:
                nonlocal ans
                ans += x
                return
            dfs(node.left, x)
            dfs(node.right, x)
        dfs(root, 0)
        return ans
```

通过 108 / 108 个通过的测试用例

image.png

用时：25min左右

M24729: 括号嵌套树

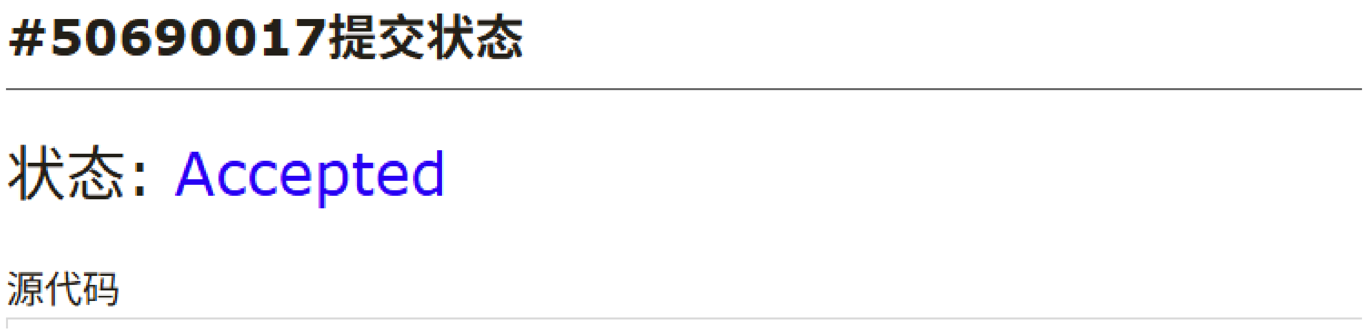
dfs, stack, http://cs101.openjudge.cn/practice/24729/

思路：

代码

```python
def solve():
    s = input().strip()
    tree = {}
    stack = []
    current_node = None
    i = 0
    while i < len(s):
        if s[i].isalpha():
            node = s[i]
            tree[node] = []
            if stack:
                tree[stack[-1]].append(node)
            current_node = node
            i += 1
        elif s[i] == '(':
            stack.append(current_node)
            i += 1
        elif s[i] == ')':
            stack.pop()
            i += 1
        elif s[i] == ',':
            i += 1

    children_set = set()
    for children in tree.values():
        children_set.update(children)

    root = None
    for node in tree:
        if node not in children_set:
            root = node
            break

    def preorder(node):
        result = [node]
        for child in tree[node]:
            result.extend(preorder(child))
        return result

    def postorder(node):
        result = []
        for child in tree[node]:
            result.extend(postorder(child))
        result.append(node)
        return result
    pre = ''.join(preorder(root))
    post = ''.join(postorder(root))
    print(pre)
    print(post)

if __name__ == "__main__":
    solve()
```

代码运行截图（至少包含有"Accepted"）



#50690017提交状态

状态: Accepted

源代码

image-1.png

用时：30min左右

T02775: 文件结构“图”

tree, http://cs101.openjudge.cn/practice/02775/

思路：

代码：

```

```

代码运行截图 （至少包含有"Accepted"）

2. 学习总结和个人收获

树的各种表示方法的基本功很重要，各种题目很有模块化的感觉，一块块函数拼起来。