

Assignment #8: 🌲 (2/3)

Updated 2223 GMT+8 Oct 27, 2025

2025 fall, Compiled by 杨浩、化院

1. 1. 题目

1.1 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:

- 二分

代码:

```

1  class Solution:
2      def sortedArrayToBST(self, nums: List[int]) ->
Optional[TreeNode]:
3          def gouzao(alist):
4              if alist:
5                  mid=(len(alist)-1)//2
6
7              pr=TreeNode(alist[mid],gouzao(alist[:mid]),gouzao(alist[mid+
8  1:]))
9              return pr
10             else:
11                 return None
12             return gouzao(nums)

```

Fence 1

代码运行截图 (至少包含有"Accepted")

通过 31 / 31 个通过的测试用例

AND-Y 提交于 2025.10.20 20:31

官方题解

写题解

⌚ 执行用时分布

ⓘ

🔥 消耗内存分布

0 ms | 击败 100.00% 🏆

18.76 MB | 击败 47.75%

📊 复杂度分析

50%

Figure 1

1.2 M07161: 森林的带度数层次序列存储

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

思路:

一开始第16行的 `while cnt==0 and alist:` 写成了 `if cnt==0:`, WA不了, 把 `build_tree()` 换了一个写法就AC, 找了半天才看到 `while` 写成 `if` 了。

代码:

```

1  from collections import deque
2  class TreeNode:
3      def __init__(self, val=None):
4          self.val=val
5          self.children=[]
6  def build_tree():
7      root=TreeNode(l[0])
8      cnt=int(l[1])
9      alist=deque([(root, cnt)])
10     pr, cnt=alist.popleft()
11     for j in range(2, len(l), 2):
12         alist.append((TreeNode(l[j]), int(l[j + 1])))
13         if cnt>0:
14             cnt-=1
15             pr.children.append(alist[-1][0])
16         while cnt==0 and alist:
17             pr, cnt=alist.popleft()
18
19     return root
20
21     '''换了一种写法就AC了...
22     def build_tree():
23         root = TreeNode(l[0])
24         alist = deque([(root, int(l[1]))])
25         j = 2
26
27         while j<len(l):
28             pr, cnt=alist[0]
29             if cnt==0:
30                 alist.popleft()
31                 continue
32             pr.children.append(TreeNode(l[j]))
33             if int(l[j+1])> 0:
34                 alist.append((pr.children[-1], int(l[j + 1])))
35             alist[0]=(pr, cnt-1)
36             j += 2
37
38         return root
39     ...
40     def postorder(root):
41         if root:

```

```

42         for child in root.children:
43             postorder(child)
44         res.append(root.val)
45     n=int(input())
46     res=[]
47     for i in range(n):
48         l=list(input().split())
49         root = build_tree()
50         postorder(root)
51     print(' '.join(res))

```

Fence 2

代码运行截图 (至少包含有"Accepted")

1.3 M27928: 遍历树

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

思路:

- 递归遍历即可

代码:

```

1  class TreeNode:
2      def __init__(self, x):
3          self.val = x
4          self.child=[]
5          self.parent=None
6
7  def __out__print(pr):
8      blist=[pr]
9      blist+=pr.child
10     blist.sort(key=lambda x:x.val)
11     for i in blist:
12         if i!=pr:
13             __out__print(i)
14         else:
15             print(i.val)
16     n=int(input())
17     pr_dic={}
18     for i in range(n):
19         alist=list(map(int,input().split()))
20         if alist[0] in pr_dic:
21             pr=pr_dic[alist[0]]
22         else:
23             pr=TreeNode(alist[0])

```

```

24         pr_dic[alist[0]]=pr
25         for j in alist[1:]:
26             if j not in pr_dic:
27                 pr_dic[j]=TreeNode(j)
28                 pr.child.append(pr_dic[j])
29                 pr_dic[j].parent=pr
30         root=pr_dic[alist[0]]
31         while root.parent:
32             root=root.parent
33         __out__print(root)

```

Fence 3

代码运行截图 (至少包含有"Accepted")

#50585442提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```

class TreeNode:
    def __init__(self, x):
        self.val = x
        self.child=[]
        self.parent=None

def __out__print(pr):
    blist=[pr]

```

基本信息

```

#: 50585442
题目: 27928
提交人: 25n2400011769
内存: 3748kB
时间: 23ms
语言: Python3
提交时间: 2025-10-27 15:29:11

```

Figure 2

1.4 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:

- dfs

代码

```

1 class Solution:
2     def sumNumbers(self, root: Optional[TreeNode]) -> int:
3         def dfs(pr,path):
4             if pr:
5                 path.append(pr.val)
6                 if pr.left:
7                     dfs(pr.left,path)
8                 if pr.right:
9                     dfs(pr.right,path)
10                if not pr.left and not pr.right:
11                    res.append(int(''.join(list(map(str,path)))))
12                    path.pop()
13            res=[]
14            dfs(root, [])

```

```
15 |         return sum(res)
```

Fence 4

代码运行截图 (至少包含有"Accepted")

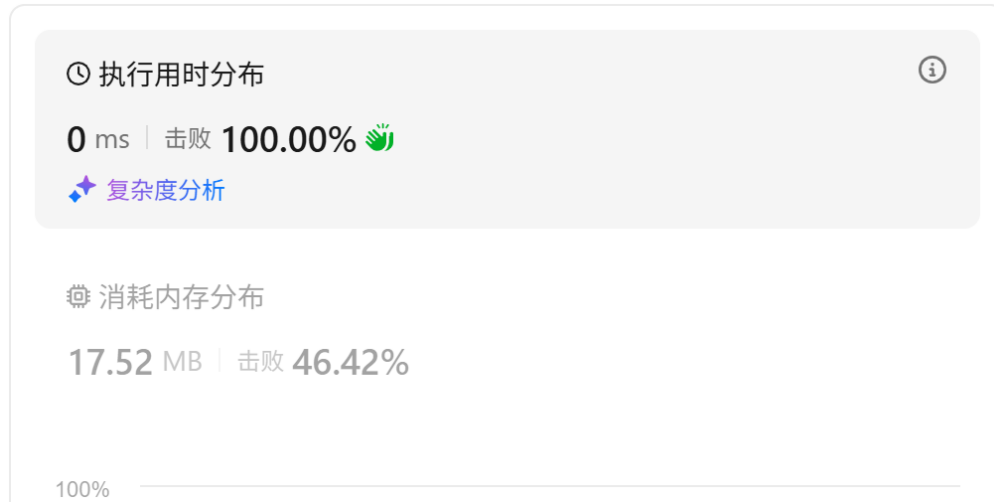


Figure 3

1.5 M24729: 括号嵌套树

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

思路:

代码

```

1  class TreeNode:
2      def __init__(self, x):
3          self.val = x
4          self.child=[]
5
6  def preorder(root):
7      if root:
8          pre_res.append(root.val)
9          for i in root.child:
10             preorder(i)
11 def inorder(root):
12     if root:
13         for i in root.child:
14             inorder(i)
15         in_res.append(root.val)
16
17 data=input()
18 root=TreeNode(data[0])
19 stack=[]
20 pr=root

```

```

21     for i in data[1:]:
22         if i=='(':
23             stack.append(pr)
24         elif i==')':
25             stack.pop()
26         elif i==',':
27             pass
28         else:
29             stack[-1].child.append(TreeNode(i))
30             pr=stack[-1].child[-1]
31     pre_res=[]
32     preorder(root)
33     print(''.join(pre_res))
34     in_res=[]
35     inorder(root)
36     print(''.join(in_res))

```

Fence 5

代码运行截图 (至少包含有"Accepted")

#50585842提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: Accepted

源代码

```

class TreeNode:
    def __init__(self, x):
        self.val = x
        self.child=[]

def preorder(root):
    if root:

```

基本信息

#: 50585842
 题目: 24729
 提交人: 25n2400011769
 内存: 3628kB
 时间: 26ms
 语言: Python3
 提交时间: 2025-10-27 15:46:11

Figure 4

1.6 T02775: 文件结构“图”

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

思路:

代码:

```

1     class TreeNode:
2         def __init__(self, name:str, deep:int):
3             self.name=name
4             self.dir__children=[]
5             self.file__children=[]
6             self.deep=deep
7         def __out__(self):
8             predix='|' * self.deep
9             print(predix+self.name)
10            for i in self.dir__children:
11                i.__out__()

```

```

12         self.file__children.sort()
13         for i in self.file__children:
14             print(predix+i)
15
16     cnt=0
17     root=TreeNode('ROOT',0)
18     stack=[root]
19     while True:
20         l=input()
21         if l=='*':
22             cnt+=1
23             print(f'DATA SET {cnt}:')
24             root.__out__()
25             root=TreeNode('ROOT',0)
26             stack=[root]
27             print()
28         elif l=='#':
29             break
30         elif l==']':
31             stack.pop()
32         elif l[0]=='d':
33
34             stack[-1].dir__children.append(TreeNode(l,stack[-1].deep+1))
35             stack.append(stack[-1].dir__children[-1])
36         elif l[0]=='f':
37             stack[-1].file__children.append(l)

```

Fence 6

代码运行截图 (至少包含有"Accepted")

#50621376提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: Accepted

源代码

```

class TreeNode:
    def __init__(self, name:str, deep:int):
        self.name=name
        self.dir__children=[]
        self.file__children=[]
        self.deep=deep
    def __out__(self):
        predix='|' * self.deep

```

基本信息

#: 50621376
 题目: 02775
 提交人: 25n2400011769
 内存: 3660kB
 时间: 22ms
 语言: Python3
 提交时间: 2025-10-29 18:04:26

Figure 5

2. 2. 学习总结和个人收获

树部分的题目模板性很强，进过一定量题目训练，熟练写递归后，做起来新题目来还是比较轻松的。LeetCode上热题100里数据结构的题这周彻底做完了，计划继续练习更为复杂的树的题目。