

Assignment #8: 树为主

Updated 1704 GMT+8 Apr 8, 2025

2025 spring, Compiled by 王梓航、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

LC108.将有序数组转换为二叉树

dfs, <https://leetcode.cn/problems/convert-sorted-array-to-binary-search-tree/>

思路: 递归地选取, 每次选最中间的即可

代码:

```
class Solution:
    def sortedArrayToBST(self, nums: List[int]) -> Optional[TreeNode]:
        def f(a):
            if not a:
                return None
            n = len(a)
            m = (n-1)//2
            new_node=TreeNode(a[m])
            if n==1:
                return new_node
            left = f(a[:m])
            right = f(a[m+1:])
            new_node.left = left
            new_node.right = right
            return new_node
```

```
return f(nums)
```

代码运行截图 (至少包含有"Accepted")

1 **通过** Python3 2 ms 19 MB
2025.03.30

M27928:遍历树

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

思路：直接dfs处理

代码：

```
n = int(input())
a = {}
h,g = set(), set()
for _ in range(n):
    b = list(map(int, input().split()))
    c = b[0]
    d = sorted(b[1:])
    a[c] = d
    h.add(c)
    g |= set(d)
root = int(list(h-g)[0])
def ans(num):
    flag = False
    for index in a[num]:
        if index > num and not flag:
            print(num)
            flag = True
    ans(index)
    if not flag:
        print(num)
ans(root)
```

代码运行截图 (至少包含有"Accepted")

状态: Accepted

源代码

```
n = int(input())
a = {}
h,g =set(), set()
for _ in range(n):
    b = list(map(int,input().split()))
    c = b[0]
    d = sorted(b[1:])
    a[c]=d
    h.add(c)
    g|=set(d)
```

基本信息

#: 48906410
题目: 27928
提交人: 24n2400011481
内存: 3680kB
时间: 22ms
语言: Python3
提交时间: 2025-04-14 11:16:01

LC129.求根节点到叶节点数字之和

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


思路: 正常处理即可, 计算出每个数应该乘的倍数


代码:

```
class Solution:
    def sumNumbers(self, root: Optional[TreeNode]) -> int:
        def f(root):
            num,ans = 0,0
            if root.left:
                x,y=f(root.left)
                num+=10*x
                ans+=y
            if root.right:
                x,y=f(root.right)
                num+=10*x
                ans+=y
            if not num:
                num = 1
            ans+=num*root.val
            return num,ans
        _,ans = f(root)
        return ans
```

代码运行截图 (至少包含有"Accepted")

通过 108 / 108 个通过的测试用例

 Hungry NorthcuttqqE 提交于 2025.04.14 12:48

 官方题解

 写题解

M22158:根据二叉树前中序序列建树

tree, <http://cs101.openjudge.cn/practice/22158/>

思路：分成三部分，根节点，左右子节点所对应部分，分开处理即可，也是不需要树，可以简单处理。

代码：

```
def f(a,b):
    if not a:
        return None
    c = a[0]
    j = b.index(c)
    f(a[1:j+1],b[:j])
    f(a[j+1:],b[j+1:])
    print(c,end = '')
while True:
    try:
        a,b = input(),input()
        f(a,b)
        print('')
    except EOFError:
        break
```

代码运行截图 (至少包含有"Accepted")

状态: Accepted

源代码

```
def f(a,b):
    if not a:
        return None
    c = a[0]
    j = b.index(c)
    f(a[1:j+1],b[:j])
    f(a[j+1:],b[j+1:])
    print(c,end = '')
while True:
```

基本信息

#: 48907492
题目: 22158
提交人: 24n2400011481
内存: 3584kB
时间: 22ms
语言: Python3
提交时间: 2025-04-14 13:33:01

M24729:括号嵌套树

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

思路：正常处理即可

代码：

```
class TreeNode:
    def __init__(self, val=0, next=None, down=None):
```

```

        self.val=val
        self.next=next
        self.down=down
a = input()
def f(a):
    if not a or a[0] in {'(', ')', ',', ' '}:
        return None
    new_node = TreeNode(a[0])
    if len(a)==1:
        return new_node
    if a[1]=='(':
        count=0
        for j in range(1,len(a)):
            if a[j]=='(':
                count+=1
            elif a[j]==')':
                count-=1
            if count==0:
                break
        left = f(a[2:j])
        right = f(a[j+2:])
    else:
        left = None
        right = f(a[2:])
    new_node.down = left
    new_node.next = right
    return new_node
s = f(a)
b,c = [],[]
def g(root):
    b.append(root.val)
    if root.down:
        g(root.down)
    if root.next:
        g(root.next)
g(s)
def h(root):
    cur = root.down
    while cur:
        h(cur)
        cur = cur.next
    c.append(root.val)
h(s)
print(''.join(b))
print(''.join(c))

```

代码运行截图 (至少包含有"Accepted")

状态: Accepted

源代码

```
class TreeNode:
    def __init__(self, val=0, next=None, down=None):
        self.val=val
        self.next=next
        self.down=down
a = input()
def f(a):
    if not a or a[0] in {'(',')','(',')'}:
        return None
```

基本信息

#: 48764322
题目: 24729
提交人: 24n2400011481
内存: 3672kB
时间: 23ms
语言: Python3
提交时间: 2025-03-30 08:58:25

LC3510.移除最小数对使数组有序II

doubly-linked list + heap, <https://leetcode.cn/problems/minimum-pair-removal-to-sort-array-ii/>

思路: 最后还是用的题解

代码:

```
class Solution:
    def minimumPairRemoval(self, nums: List[int]) -> int:
        s1 = SortedList() # (相邻元素和, 左边那个数的下标)
        idx = SortedList(range(len(nums))) # 剩余下标
        dec = 0 # 递减的相邻对的个数

        for i, (x, y) in enumerate(pairwise(nums)):
            if x > y:
                dec += 1
                s1.add((x + y, i))

        ans = 0
        while dec > 0:
            ans += 1

            s, i = s1.pop(0) # 删除相邻元素和最小的一对
            k = idx.bisect_left(i)

            # (当前元素, 下一个数)
            nxt = idx[k + 1]
            if nums[i] > nums[nxt]: # 旧数据
                dec -= 1

            # (前一个数, 当前元素)
            if k > 0:
                pre = idx[k - 1]
                if nums[pre] > nums[i]: # 旧数据
                    dec -= 1
                if nums[pre] > s: # 新数据
                    dec += 1
                s1.remove((nums[pre] + nums[i], pre))
```

```

        sl.add((nums[pre] + s, pre))

        # (下一个数, 下下一个数)
        if k + 2 < len(idx):
            nxt2 = idx[k + 2]
            if nums[nxt] > nums[nxt2]: # 旧数据
                dec -= 1
            if s > nums[nxt2]: # 新数据 (当前元素, 下下一个数)
                dec += 1
            sl.remove((nums[nxt] + nums[nxt2], nxt))
            sl.add((s + nums[nxt2], i))

        nums[i] = s # 把 nums[nxt] 加到 nums[i] 中
        idx.remove(nxt) # 删除 nxt

    return ans

```

作者: 灵茶山艾府


链接: <https://leetcode.cn/problems/minimum-pair-removal-to-sort-array-ii/solutions/3641787/mo-ni-liang-ge-you-xu-ji-he-by-endlessch-gx8s/>

来源: 力扣 (LeetCode)

著作权归作者所有。商业转载请联系作者获得授权, 非商业转载请注明出处。

代码运行截图 (至少包含有"Accepted")

通过 680 / 680 个通过的测试用例

 Hungry NorthcuttqqE 提交于 2025.04.14 23:14

 写题解

2. 学习总结和收获

如果发现作业题目相对简单, 有否寻找额外的练习题目, 如“数算2025spring每日选做”、LeetCode、Codeforces、洛谷等网站上的题目。

最近因为期中考试, 做题比较少, 感觉有点吃力, 接下来的时间会尽量跟上。