

Assignment #9: Huffman, BST & Heap

Updated 1834 GMT+8 Apr 15, 2025

2025 spring, Compiled by <mark>汤伟杰，信息管理系</mark>

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

LC222.完全二叉树的节点个数

dfs, <https://leetcode.cn/problems/count-complete-tree-nodes/>

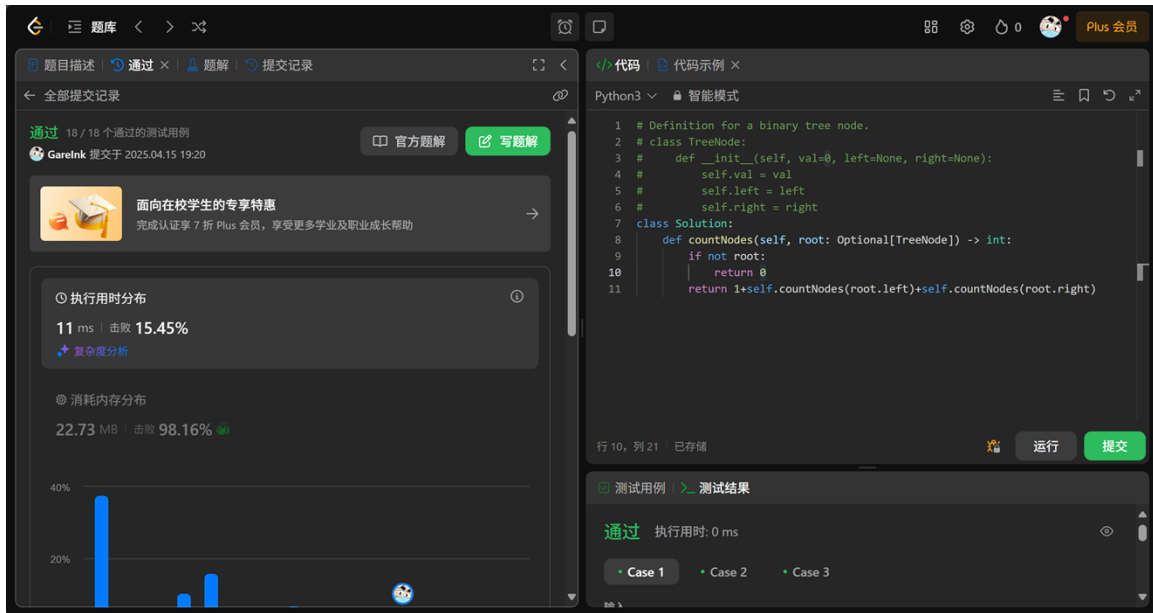
思路：

递归，

代码：

```
# 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 countNodes(self, root: Optional[TreeNode]) -> int:
        if not root:
            return 0
        return 1+self.countNodes(root.left)+self.countNodes(root.right)
```

代码运行截图 <mark>（至少包含有"Accepted"）</mark>



LC103.二叉树的锯齿形层序遍历

bfs, <https://leetcode.cn/problems/binary-tree-zigzag-level-order-traversal/>

思路:

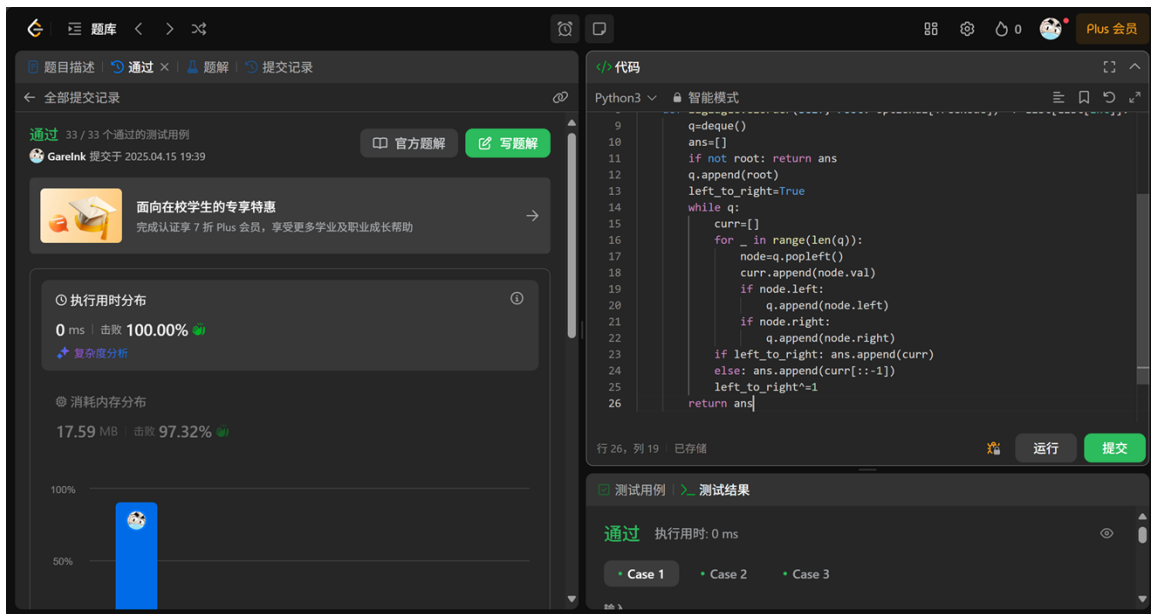
只需要正常层序遍历, 多加一个布尔遍历来判断当前层次是正序加入答案还是逆序加入答案即可。

代码:

```
# 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 zigzagLevelOrder(self, root: Optional[TreeNode]) -> List[List[int]]:
        q=deque()
        ans=[]
        if not root: return ans
        q.append(root)
        left_to_right=True
        while q:
            curr=[]
            for _ in range(len(q)):
                node=q.popleft()
                curr.append(node.val)
                if node.left:
                    q.append(node.left)
                if node.right:
                    q.append(node.right)
            if left_to_right: ans.append(curr)
            else: ans.append(curr[::-1])
            left_to_right = not left_to_right
```

```
left_to_right ^= 1
return ans
```

代码运行截图 <mark>（至少包含有"Accepted"） </mark>



M04080:Huffman 编码树

greedy, <http://cs101.openjudge.cn/practice/04080/>

思路:

把每个数字本身的数值看作文本的字符 和 字符的出现频次来建树，树的根节点保证 val 值都是 None 以方便后续的递归找到叶子节点，最后的计算运用递归，设置全局变量 ans，向左递归写一次，向右递归写一次，退出条件为当前节点的 val 值不是 None，那就加到 ans 上。递归的时候把 height 带上，方便。

代码:

```
#pylint:skip-file
from heapq import heappop,heappush,heapify

class Node:
    def __init__(self,fre,val):
        self.fre=fre
        self.val=val
        self.left=None
        self.right=None

    def __lt__(self, other):
        return self.fre<other.fre

def build(s):
    h=[Node(s[i],s[i]) for i in range(len(s))] #(fre,val)
    heapify(h)
```

```

while len(h)>1:
    left=heappop(h)
    right=heappop(h)
    merged=Node(left.fre+right.fre,None)
    merged.left=left
    merged.right=right
    heappush(h,merged)

return h[0]

def cal(root,height):
    global ans
    if root.val is None:
        cal(root.left,height+1)
        cal(root.right,height+1)
        return
    ans+=root.val*height
    return

n=int(input())
s=list(map(int,input().split()))
ans=0
root=build(s)
cal(root,0)
print(ans)

```

代码运行截图 <mark>（至少包含有"Accepted"）</mark>

OpenJudge 题目ID, 标题, 描述 24n2400016635 信箱 账号

CS101 / 题库 (包括计概、数算题目)

题目 排名 状态 提问

#48919164提交状态 查看 提交 统计 提问

状态: Accepted

源代码

```

'''
    0
  0 0 5
0 0 3
1 1
'''
#pylint:skip-file

from heapq import heappop, heappush, heapify

class Node:
    def __init__(self, fre, val):
        self.fre=fre
        self.val=val
        self.left=None
        self.right=None

    def __lt__(self, other):
        return self.fre<other.fre

def build(s):

```

基本信息

#:	48919164
题目:	04080
提交人:	24n2400016635
内存:	3688kB
时间:	22ms
语言:	Python3
提交时间:	2025-04-15 19:42:37

M05455: 二叉搜索树的层次遍历

<http://cs101.openjudge.cn/practice/05455/>

思路:

对于每个数字，作为新节点插入 **bst**，从根节点开始递归比较大小。然后就是层次遍历。

代码：

```
from collections import deque

class TreeNode:
    def __init__(self, val=None):
        self.val=val
        self.left=None
        self.right=None

def buildTree(root,i):
    if not root:
        return TreeNode(i)
    else:
        if i<root.val:
            root.left=buildTree(root.left,i)
        elif i>root.val:
            root.right=buildTree(root.right,i)
    return root

def level_traversal(root):
    q=deque([root])
    ans=[root.val]
    while q:
        node=q.popleft()
        if node.left:
            q.append(node.left)
            ans.append(node.left.val)
        if node.right:
            q.append(node.right)
            ans.append(node.right.val)
    return ans

s=list(map(int,input().split()))
root=None
for i in s: root=buildTree(root,i)
print(*level_traversal(root))
```

代码运行截图 <mark>（至少包含有"Accepted"） </mark>

OpenJudge 题目ID, 标题, 描述 24n2400016635 信箱 账号

CS101 / 题库 (包括计概、数算题目)

题目 排名 状态 提问

#48919411提交状态 查看 提交 统计 提问

状态: Accepted

源代码

```
from collections import deque

class TreeNode:
    def __init__(self, val=None):
        self.val=val
        self.left=None
        self.right=None

def buildTree(root,i):
    if not root:
        return TreeNode(i)
    else:
        if i<root.val:
            root.left=buildTree(root.left,i)
        elif i>root.val:
            root.right=buildTree(root.right,i)
        return root

def level_traversal(root):
    q=deque([root])
    ans=[root.val]
    while q:
```

基本信息

#: 48919411
题目: 05455
提交人: 24n2400016635
内存: 3644kB
时间: 20ms
语言: Python3
提交时间: 2025-04-15 20:07:55

M04078: 实现堆结构

手搓实现, <http://cs101.openjudge.cn/practice/04078/>

类似的题目是 晴问 9.7: 向下调整构建大顶堆, <https://sunnywhy.com/sfbj/9/7>

思路:

手搓思路来源于课件, 寒假的时候有研读并抄写过, 自己写感觉还需要 debug 半天。类的主体仍然是 list 列表, python 真的太强大了(哈哈), 每次加入就 append 进去, 然后把最后一个位置的新元素向上调整; 弹出就先把列表的第一个和最后一个位置互换, 然后弹出最后一个位置的元素, 再对第一个位置的元素向下调整。父节点的索引是 $i//2$, 子节点的索引是 $i*2$ 和 $i*2+1$, 为了维护父节点的值比子节点的值小, 向上调整只需要查看父节点是不是比子节点大, 大了就换位置; 向下调整则需要找到子节点的较小者, 查看这个较小者是不是比父节点小? 小了就互换。手搓, 爽! 但是代码好长好容易错。。

代码:

```
class Heap:
    def __init__(self):
        self.heap=[0]
        self.size=0

    def append(self,val):
        self.heap.append(val)
        self.size+=1
        self.adjustUp(self.size)

    def adjustUp(self,i):
        while i>1 and self.heap[i]<self.heap[i//2]:
            self.heap[i],self.heap[i//2]=self.heap[i//2],self.heap[i]
            i//=2

    def pop(self):
```

```

        self.heap[1],self.heap[-1]=self.heap[-1],self.heap[1]
        returnd=self.heap.pop()
        self.size-=1
        self.adjustDown(1)
        return returnd

    def adjustDown(self,i):
        while i*2<=self.size:
            if i*2+1>self.size:
                minChild_idx=i*2
            else:
                minChild_idx=min(i*2,i*2+1,key=lambda x:self.heap[x])

            if self.heap[i]>self.heap[minChild_idx]:

self.heap[i],self.heap[minChild_idx]=self.heap[minChild_idx],self.heap[i]
            i=minChild_idx

h=Heap()
n=int(input())
for _ in range(n):
    y,*u=map(int,input().split())
    if y==1:
        h.append(u[0])
    else:
        print(h.pop())

```

代码运行截图 <mark>（至少包含有"Accepted"） </mark>

OpenJudge
题目ID, 标题, 描述
24n2400016635
信箱
账号

CS101 / 题库（包括计概、数学题目）
题目
排名
状态
提问

#48920237提交状态
查看
提交
统计
提问

状态: Accepted

源代码

```

class Heap:
    def __init__(self):
        self.heap=[0]
        self.size=0

    def append(self,val):
        self.heap.append(val)
        self.size+=1
        self.adjustUp(self.size)

    def adjustUp(self,i):
        while i>1 and self.heap[i]<self.heap[i//2]:
            self.heap[i],self.heap[i//2]=self.heap[i//2],self.heap[i]
            i//=2

    def pop(self):
        self.heap[1],self.heap[-1]=self.heap[-1],self.heap[1]
        returnd=self.heap.pop()
        self.size-=1
        self.adjustDown(1)
        return returnd

```

基本信息

#: 48920237
题目: 04078
提交人: 24n2400016635
内存: 4072kB
时间: 924ms
语言: Python3
提交时间: 2025-04-15 21:09:34

T22161: 哈夫曼编码树

greedy, <http://cs101.openjudge.cn/practice/22161/>

思路:

思路来自寒假（，主要是建完树之后获取编码的 `code_book`（感觉 `book` 这个词很形象，用字典存，每次都是查书看看有没有这个词条，有就加到答案里面），再反转键值对，就得到了编码和解码的字典。

代码:

```
from heapq import heappop, heappush, heapify
class Node:
    def __init__(self, fre, chr):
        self.fre = fre
        self.chr = chr
        self.left = None
        self.right = None
    def __lt__(self, other):
        return self.fre < other.fre

def build(fre_chr):
    h = [Node(fre, chr) for chr, fre in fre_chr.items()]
    heapify(h)

    while len(h) > 1:
        l = heappop(h)
        r = heappop(h)
        merged = Node(l.fre + r.fre, min(l.chr, r.chr))
        merged.left = l
        merged.right = r
        heappush(h, merged)

    return h[0]

def get_code(root, curr, code_book):
    if root.left == root.right: #leave
        code_book[root.chr] = curr
        return
    get_code(root.left, curr + '0', code_book)
    get_code(root.right, curr + '1', code_book)
    return

n = int(input())
fre_chr = {}
for _ in range(n):
    c, f = input().split()
    fre_chr[c] = int(f)
root = build(fre_chr)
code_book = {}
get_code(root, '', code_book)
```



```

reversed_code={v:k for k,v in code_book.items()}
while True:
    try:
        l=input()
        if l[0].isalpha():
            ans=''
            for i in l:
                ans+=code_book[i]
            print(ans)
        else:
            ans=''
            buffer=''
            for bit in l:
                buffer+=bit
                if buffer in reversed_code:
                    ans+=reversed_code[buffer]
                    buffer=''
            print(ans)
    except EOFError:
        break

```

代码运行截图 <mark>（至少包含有"Accepted"）</mark>



OpenJudge 题目ID, 标题, 描述 24n2400016635 信箱 账号

CS101 / 题库 (包括计概、数算题目)

题目 排名 状态 提问

#48920926提交状态 查看 提交 统计 提问

状态: Accepted

源代码

```

from heapq import heappop, heappush, heapify
class Node:
    def __init__(self, fre, chr):
        self.fre=fre
        self.chr=chr
        self.left=None
        self.right=None
    def __lt__(self, other):
        return self.fre<other.fre

def build(fre_chr):
    h=[Node(fre,chr) for chr,fre in fre_chr.items()]
    heapify(h)

    while len(h)>1:
        l=heappop(h)
        r=heappop(h)
        merged=Node(l.fre+r.fre, min(l.chr, r.chr))
        merged.left=l
        merged.right=r
        heappush(h,merged)

```

基本信息

#:	48920926
题目:	22161
提交人:	24n2400016635
内存:	3652kB
时间:	23ms
语言:	Python3
提交时间:	2025-04-15 22:14:40

2. 学习总结和收获

<mark>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。</mark>

每日选做还在跟进，课件还在继续学习，继续加油！这次作业比上次的感觉要好做一点，估计是因为模板性很强。不过计概时候的题目比如 dp 之类的好像都有点生了，需要复习复习了！！