Assignment #9: Huffman, BST & Heap

Updated 1834 GMT+8 Apr 15, 2025

2025 spring, Complied by 郑涵予 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

LC222.完全二叉树的节点个数

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

思路：

这题可以使用二分法，先找出最大层数然后就开始二分。（用时约15min）

代码：

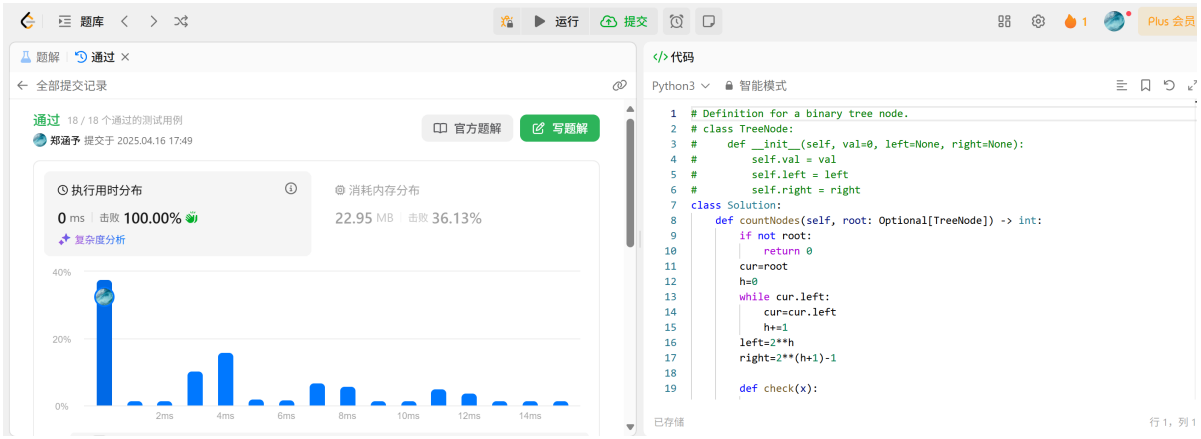
```
# Definition for a binary tree node.
# class TreeNode:
#     def __init__(self, val=0, left=None, right=None):
#         self.val = val
#         self.left = left
#         self.right = right
class Solution:
```

```python
    def countNodes(self, root: Optional[TreeNode]) -> int:
        if not root:
            return 0
        cur=root
        h=0
        while cur.left:
            cur=cur.left
            h+=1
        left=2**h
        right=2**(h+1)-1

        def check(x):
            path=[]
            while x!=1:
                if x&1:
                    path.append(1)
                else:
                    path.append(0)
                x//=2
            cur=root
            for i in range(len(path)-1,-1,-1):
                if not cur:
                    return False
                if path[i]==1:
                    cur=cur.right
                else:
                    cur=cur.left
            if not cur:
                return False
            return True

        while left<right:
            mid=(left+right+1)//2
            if check(mid):
                left=mid
            else:
                right=mid-1
        return left
```

代码运行截图 （至少包含有"Accepted"）



LC103.二叉树的锯齿形层序遍历

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

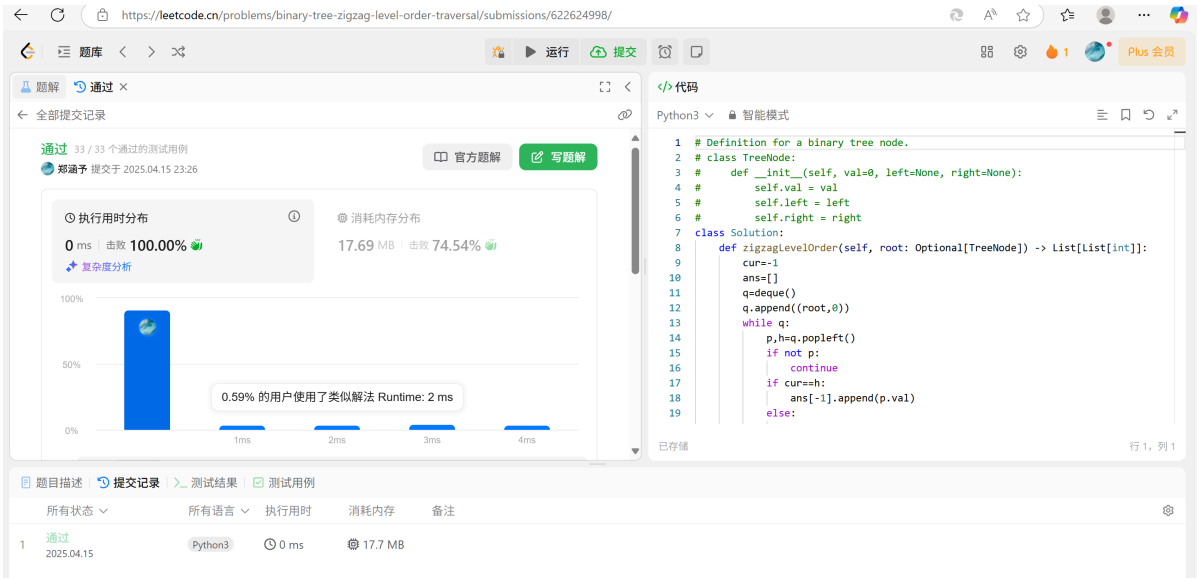
思路：

直接层序历遍，然后把奇数层翻转一下就行（用时约10min)

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]]:
        cur=-1
        ans=[]
        q=deque()
        q.append((root,0))
        while q:
            p,h=q.popleft()
            if not p:
                continue
            if cur==h:
                ans[-1].append(p.val)
            else:
                cur+=1
                ans.append([p.val])
            q.append((p.left,h+1))
            q.append((p.right,h+1))
        for i in range(1,len(ans),2):
            ans[i]=ans[i][::-1]
        return ans
```

代码运行截图 （至少包含有"Accepted"）



M04080:Huffman编码树

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

思路：

本质上就是贪心，这题无需建树，所以代码量很小（用时约3min)

代码：

```
import heapq
n=int(input())
a=list(map(int,input().split()))
q=[]
for x in a:
    heapq.heappush(q,x)
ans=0
while len(q)>1:
    x,y=heapq.heappop(q),heapq.heappop(q)
    ans=ans+x+y
    heapq.heappush(q,x+y)
print(ans)
```

代码运行截图 （至少包含有"Accepted"）

#48928453提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```
import heapq
n=int(input())
a=list(map(int,input().split()))
q=[]
for x in a:
    heapq.heappush(q,x)
ans=0
while len(q)>1:
    x,y=heapq.heappop(q),heapq.heappop(q)
    ans=ans+x+y
    heapq.heappush(q,x+y)
print(ans)
```

基本信息

#: 48928453
题目: 04080
提交人: 24n2400011325
内存: 3616kB
时间: 19ms
语言: Python3
提交时间: 2025-04-16 16:56:54

English 帮助 关于

M05455: 二叉搜索树的层次遍历

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

思路:

二叉搜索树建树的模板题（用时约10min)

代码:

```
class Node():
    def __init__(self,val=0):
        self.val=val
        self.left=None
        self.right=None

def lever_order(root):
    from collections import deque
    q=deque()
    q.append(root)
    res=[]
    while q:
        p=q.popleft()
        if not p:
            continue
        res.append(p.val)
        q.append(p.left)
        q.append(p.right)
    return res

def insert(root,val):
    if val>root.val:
        if not root.right:
            root.right=Node(val)
        else:
```

```
            insert(root.right,val)
    else:
        if not root.left:
            root.left=Node(val)
        else:
            insert(root.left,val)

a=list(map(int,input().split()))
root=Node(a[0])
s=set()
s.add(a[0])
for i in range(1,len(a)):
    if a[i] in s:
        continue
    s.add(a[i])
    insert(root,a[i])
print(' '.join(map(str,lever_order(root))))
```

代码运行截图 （至少包含有"Accepted"）

#48950240提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```
class Node():
    def __init__(self,val=0):
        self.val=val
        self.left=None
        self.right=None

def lever_order(root):
    from collections import deque
    q=deque()
    q.append(root)
    res=[]
    while q:
        p=q.popleft()
        if not p:
            continue
        res.append(p.val)
        q.append(p.left)
        q.append(p.right)
    return res

def insert(root,val):
    if val>root.val:
        if not root.right:
            root.right=Node(val)
        else:
            insert(root.right,val)
```

基本信息

#: 48950240
题目: 05455
提交人: 24n2400011325
内存: 3664kB
时间: 21ms
语言: Python3
提交时间: 2025-04-18 23:11:27

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：

相当于自己手搓一个堆（用时约15min）

代码:

```
class BinaryHeapq:
    def __init__(self):
        self.q=[-1]

    def up_adjust(self,index):
        while index!=1 and self.q[index]<self.q[index//2]:
            self.q[index],self.q[index//2]=self.q[index//2],self.q[index]
            index//=2

    def down_adjust(self,index):
        while 2*index<len(self.q):
            pos=index*2 if 2*index+1>=len(self.q) or self.q[2*index]
<self.q[2*index+1] else index*2+1
            if self.q[index]>self.q[pos]:
                self.q[index],self.q[pos]=self.q[pos],self.q[index]
                index=pos
            else:
                break


    def insert(self,val):
        self.q.append(val)
        self.up_adjust(len(self.q)-1)

    def delete(self):
        res=self.q[1]
        self.q[1],self.q[-1]=self.q[-1],self.q[1]
        self.q.pop()
        self.down_adjust(1)
        return res

n=int(input())
q=BinaryHeapq()
for _ in range(n):
    s=list(map(int,input().split()))
    if len(s)==1:
        print(q.delete())
    else:
        q.insert(s[1])
```

代码运行截图 （至少包含有"Accepted"）

#48950097提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```
class BinaryHeapq:
    def __init__(self):
        self.q=[-1]

    def up_adjust(self,index):
        while index!=1 and self.q[index]<self.q[index//2]:
            self.q[index],self.q[index//2]=self.q[index//2],self.q[inde
            index//=2

    def down_adjust(self,index):
        while 2*index<len(self.q):
            pos=index*2 if 2*index+1>=len(self.q) or self.q[2*index]<se
            if self.q[index]>self.q[pos]:
                self.q[index],self.q[pos]=self.q[pos],self.q[index]
                index=pos
            else:
                break


    def insert(self,val):
        self.q.append(val)
        self.up_adjust(len(self.q)-1)

    def delete(self):
        res=self.q[1]
```

基本信息

#: 48950097
题目: 04078
提交人: 24n2400011325
内存: 4084kB
时间: 663ms
语言: Python3
提交时间: 2025-04-18 22:46:37

T22161: 哈夫曼编码树

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

思路:

这题要求建树就麻烦了不少，但本质上还是模板题（用时约15min)

代码:

```
import heapq
class Node():
    def __init__(self, char="",val=0):
        self.val=val
        self.char=char
        self.left=None
        self.right=None

    def __lt__(self,other):
        if self.val==other.val:
            return self.char<other.char
        return self.val<other.val

n=int(input())
q=[]
a=[]
for _ in range(n):
    s,x=input().split()
    a.append(x)
    x=int(x)
    heapq.heappush(q,Node(s,x))
```

```
while len(q)>1:
    x,y=heapq.heappop(q),heapq.heappop(q)
    new_node=Node(min(x.char,y.char),x.val+y.val)
    new_node.left=x
    new_node.right=y
    heapq.heappush(q,new_node)
char_to_num,num_to_char={},{}
root=q[0]

def dfs(root,path):
    if not root:
        return
    if not root.left and not root.right:
        char_to_num[root.char]=path
        num_to_char[path]=root.char
    dfs(root.left,path+'0')
    dfs(root.right,path+'1')

dfs(root,"")
while 1:
    try:
        s=input()
        if s.isdigit():
            temp=""
            pos=0
            res=""
            while pos<len(s):
                temp+=s[pos]
                if temp in num_to_char:
                    res+=num_to_char[temp]
                    temp=""
                pos+=1
            print(res)
        else:
            res=[]
            for x in s:
                res.append(char_to_num[x])
            print(''.join(res))
    except EOFError:
        break
```

代码运行截图 （至少包含有"Accepted"）

#48929036提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```
import heapq
class Node():
    def __init__(self, char="",val=0):
        self.val=val
        self.char=char
        self.left=None
        self.right=None

    def __lt__(self,other):
        if self.val==other.val:
            return self.char<other.char
        return self.val<other.val

n=int(input())
q=[]
a=[]
for _ in range(n):
    s,x=input().split()
    a.append(x)
    x=int(x)
    heapq.heappush(q,Node(s,x))
while len(q)>1:
    x,y=heapq.heappop(q),heapq.heappop(q)
```

基本信息

#: 48929036
题目: 22161
提交人: 24n2400011325
内存: 3716kB
时间: 19ms
语言: Python3
提交时间: 2025-04-16 17:16:40

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。

这周补上了之前因为期中欠下的每日选做（其实也不多，因为好多寒假做过了），感觉比较繁琐的数据结构自己实现起来还是会出现很多bug,比如AVL树就总是写不对，还是得多熟悉一下。