# Assignment #6: "树"算：Huffman,BinHeap,BST,AVL,DisjointSet

Updated 2214 GMT+8 March 24, 2024

2024 spring, Complied by 苏王捷 工学院

**说明：**

1）这次作业内容不简单，耗时长的话直接参考题解。

2）请把每个题目解题思路（可选），源码Python, 或者C++（已经在Codeforces/Openjudge上AC），截图（包含Accepted），填写到下面作业模版中（推荐使用 typora https://typoraio.cn ，或者用word）。AC 或者没有AC，都请标上每个题目大致花费时间。

3）提交时候先提交pdf文件，再把md或者doc文件上传到右侧"作业评论"。Canvas需要有同学清晰头像、提交文件有pdf、"作业评论"区有上传的md或者doc附件。

4）如果不能在截止前提交作业，请写明原因。

**编程环境**

（请改为同学的操作系统、编程环境等）

操作系统：Windows 11

Python编程环境：Spyder IDE 5.5.1

## 1. 题目

### 22275: 二叉搜索树的遍历

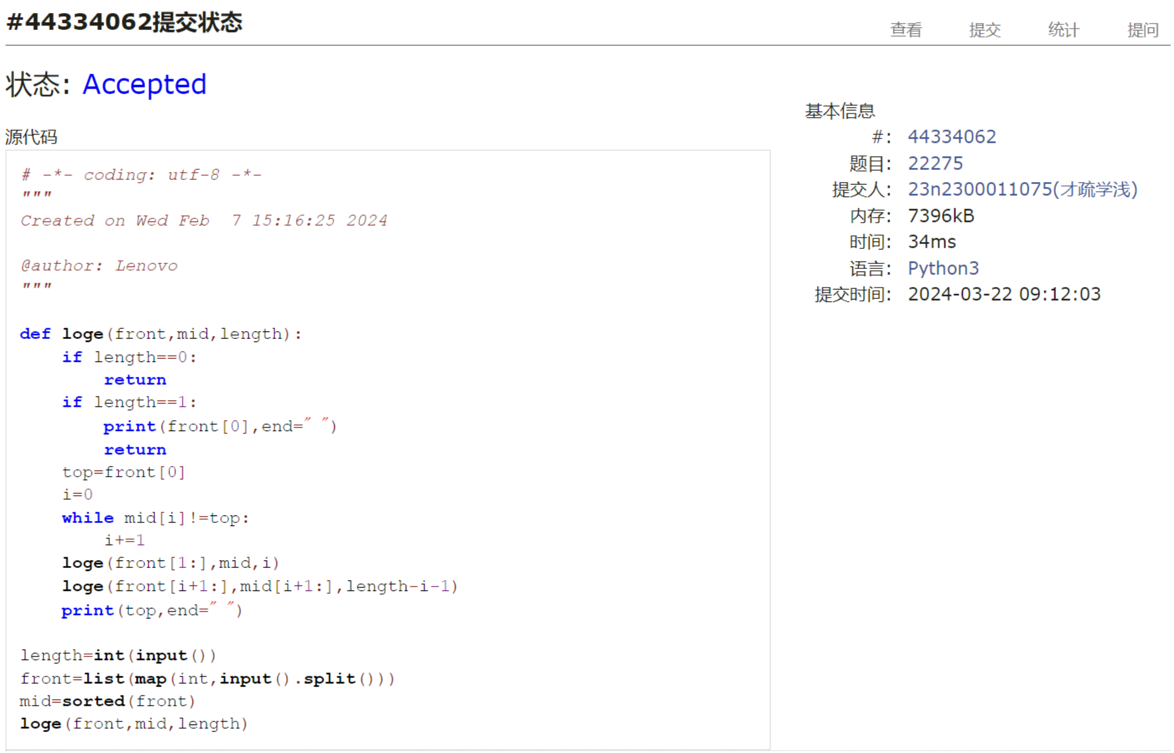
http://cs101.openjudge.cn/practice/22275/

思路：找到左子树、右子树、父节点，按序递归输出

代码

```
# # -*- coding: utf-8 -*-
"""
Created on Wed Feb  7 15:16:25 2024

@author: Lenovo
"""

def loge(front,mid,length):
    if length==0:
        return
    if length==1:
        print(front[0],end=" ")
        return
    top=front[0]
    i=0
    while mid[i]!=top:
        i+=1
    loge(front[1:],mid,i)
    loge(front[i+1:],mid[i+1:],length-i-1)
    print(top,end=" ")

length=int(input())
front=list(map(int,input().split()))
mid=sorted(front)
loge(front,mid,length)
```

代码运行截图 （至少包含有"Accepted"）



#44334062提交状态

状态: Accepted

源代码

```
# -*- coding: utf-8 -*-
"""
Created on Wed Feb  7 15:16:25 2024

@author: Lenovo
"""

def loge(front,mid,length):
    if length==0:
        return
    if length==1:
        print(front[0],end=" ")
        return
    top=front[0]
    i=0
    while mid[i]!=top:
        i+=1
    loge(front[1:],mid,i)
    loge(front[i+1:],mid[i+1:],length-i-1)
    print(top,end=" ")

length=int(input())
front=list(map(int,input().split()))
mid=sorted(front)
loge(front,mid,length)
```

基本信息

| | |
|---|---|
| #: | 44334062 |
| 题目: | 22275 |
| 提交人: | 23n2300011075(才疏学浅) |
| 内存: | 7396kB |
| 时间: | 34ms |
| 语言: | Python3 |
| 提交时间: | 2024-03-22 09:12:03 |

## 05455: 二叉搜索树的层次遍历

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

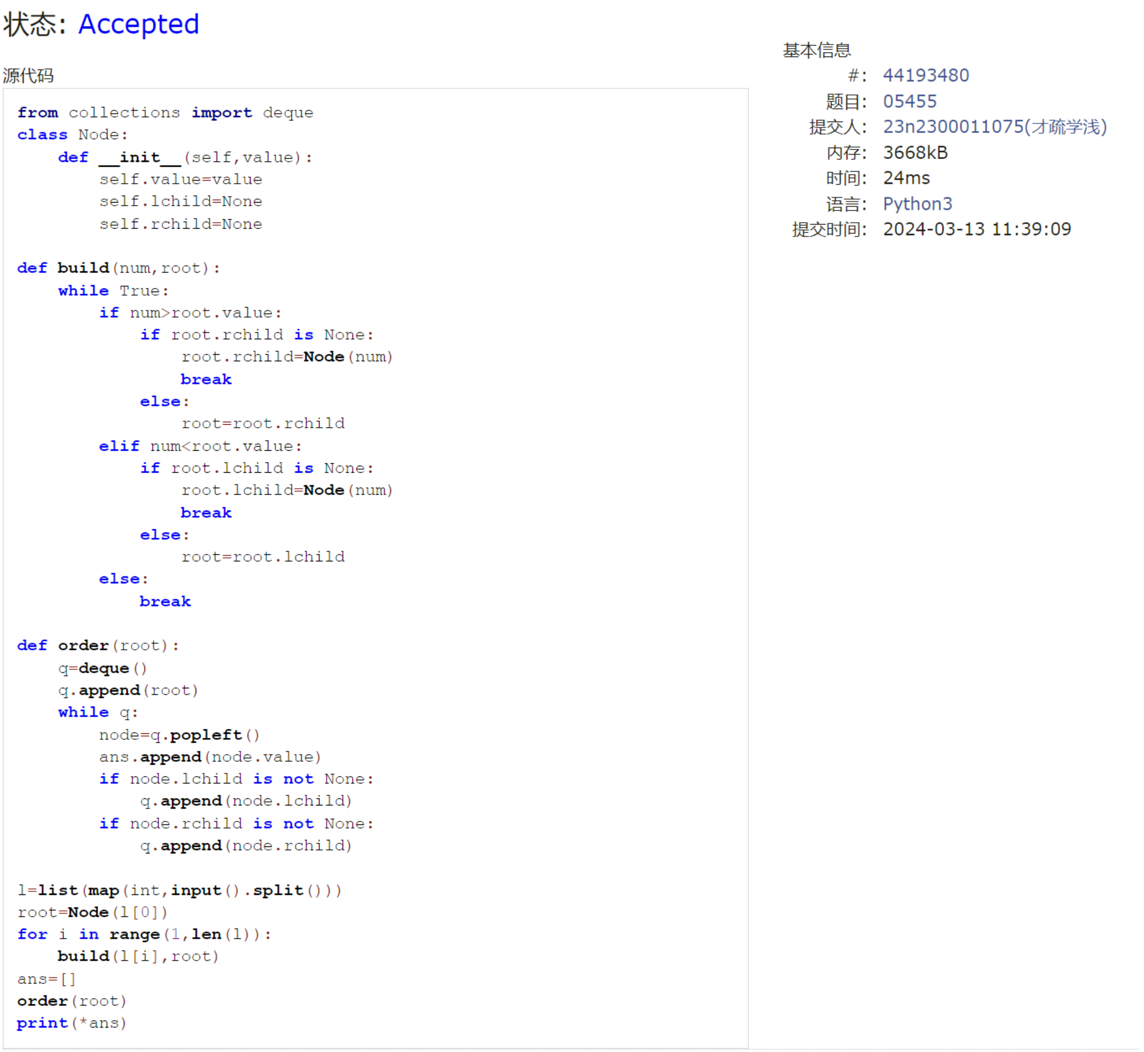
思路：用二叉搜索树性质建树，然后用deque实现层序遍历

代码

```
#
```

```
from collections import deque
class Node:
    def __init__(self,value):
        self.value=value
        self.lchild=None
        self.rchild=None

def build(num,root):
    while True:
        if num>root.value:
            if root.rchild is None:
                root.rchild=Node(num)
                break
            else:
                root=root.rchild
        elif num<root.value:
            if root.lchild is None:
                root.lchild=Node(num)
                break
            else:
                root=root.lchild
        else:
            break

def order(root):
    q=deque()
    q.append(root)
    while q:
        node=q.popleft()
        ans.append(node.value)
        if node.lchild is not None:
            q.append(node.lchild)
        if node.rchild is not None:
            q.append(node.rchild)

l=list(map(int,input().split()))
root=Node(l[0])
for i in range(1,len(l)):
    build(l[i],root)
ans=[]
order(root)
print(*ans)
```

代码运行截图 （至少包含有"Accepted"）



状态: Accepted

源代码

```
from collections import deque
class Node:
    def __init__(self,value):
        self.value=value
        self.lchild=None
        self.rchild=None

def build(num,root):
    while True:
        if num>root.value:
            if root.rchild is None:
                root.rchild=Node(num)
                break
            else:
                root=root.rchild
        elif num<root.value:
            if root.lchild is None:
                root.lchild=Node(num)
                break
            else:
                root=root.lchild
        else:
            break

def order(root):
    q=deque()
    q.append(root)
    while q:
        node=q.popleft()
        ans.append(node.value)
        if node.lchild is not None:
            q.append(node.lchild)
        if node.rchild is not None:
            q.append(node.rchild)

l=list(map(int,input().split()))
root=Node(l[0])
for i in range(1,len(l)):
    build(l[i],root)
ans=[]
order(root)
print(*ans)
```

基本信息

| | |
|---|---|
| #: | 44193480 |
| 题目: | 05455 |
| 提交人: | 23n2300011075(才疏学浅) |
| 内存: | 3668kB |
| 时间: | 24ms |
| 语言: | Python3 |
| 提交时间: | 2024-03-13 11:39:09 |

## 04078: 实现堆结构

http://cs101.openjudge.cn/practice/04078/

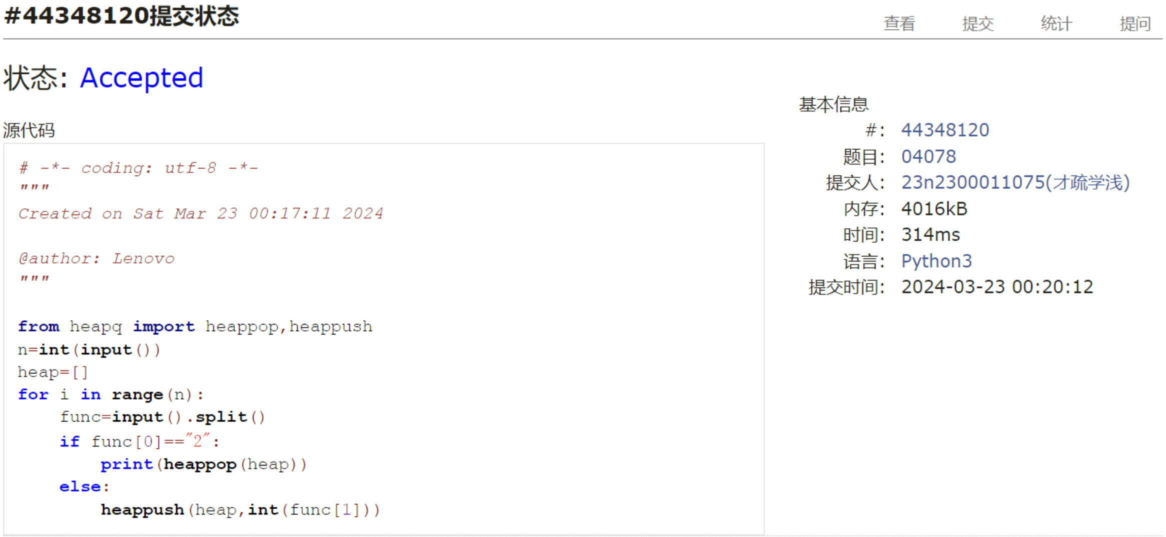
练习自己写个BinHeap。当然机考时候，如果遇到这样题目，直接import heapq。手搓栈、队列、堆、AVL等，考试前需要搓个遍。

思路：直接用heap实现堆

代码

```
# # -*- coding: utf-8 -*-
"""
Created on Sat Mar 23 00:17:11 2024

@author: Lenovo
"""

from heapq import heappop,heappush
n=int(input())
heap=[]
for i in range(n):
    func=input().split()
    if func[0]=="2":
        print(heappop(heap))
    else:
        heappush(heap,int(func[1]))
```

代码运行截图 （AC代码截图，至少包含有"Accepted"）



#44348120提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```
# -*- coding: utf-8 -*-
"""
Created on Sat Mar 23 00:17:11 2024

@author: Lenovo
"""

from heapq import heappop,heappush
n=int(input())
heap=[]
for i in range(n):
    func=input().split()
    if func[0]=="2":
        print(heappop(heap))
    else:
        heappush(heap,int(func[1]))
```

基本信息

| | |
|---|---|
| #: | 44348120 |
| 题目: | 04078 |
| 提交人: | 23n2300011075(才疏学浅) |
| 内存: | 4016kB |
| 时间: | 314ms |
| 语言: | Python3 |
| 提交时间: | 2024-03-23 00:20:12 |

## 22161: 哈夫曼编码树

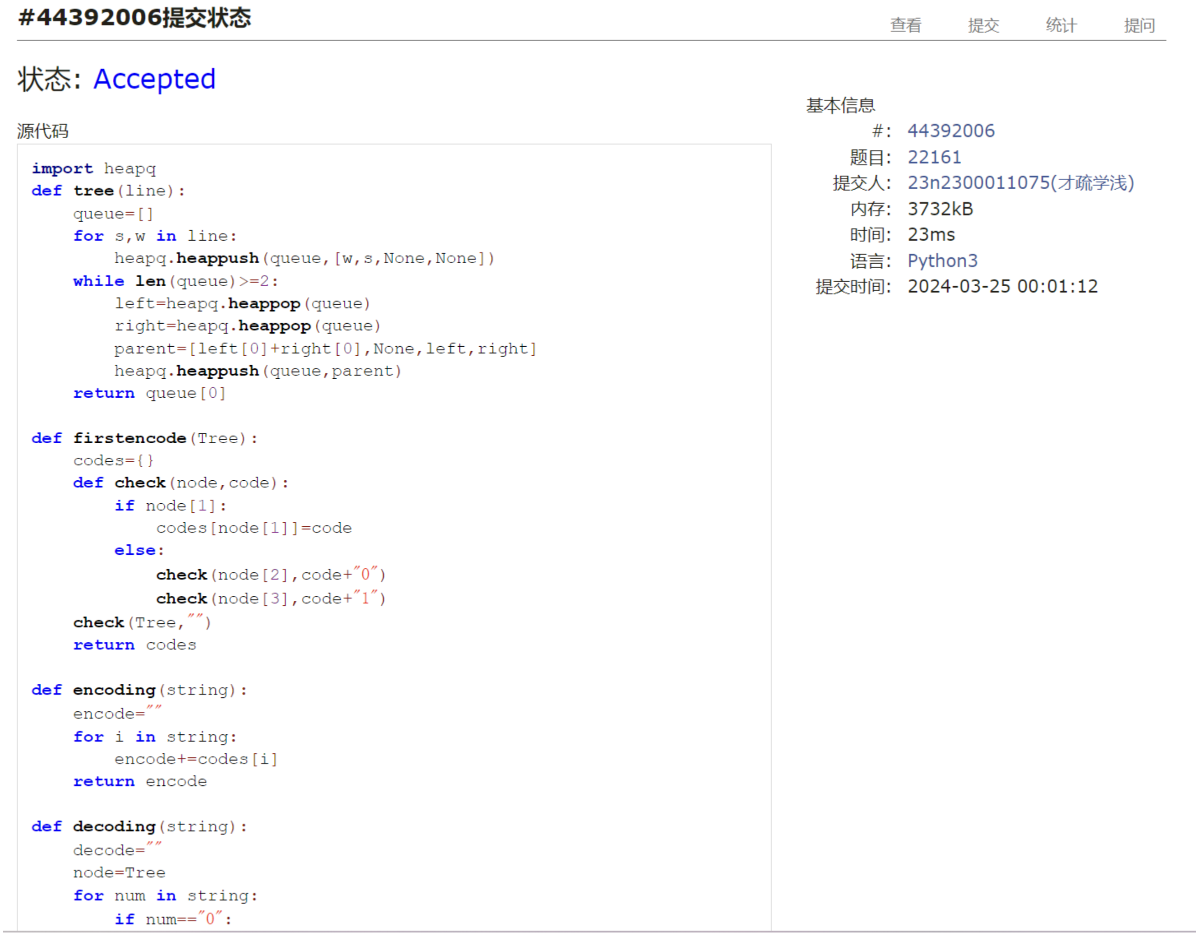
http://cs101.openjudge.cn/practice/22161/

思路：许久之前写的，用列表模拟，可以方便地用节点值和左右子树排序

代码

```
#
import heapq
def tree(line):
    queue=[]
    for s,w in line:
        heapq.heappush(queue,[w,s,None,None])
    while len(queue)>=2:
        left=heapq.heappop(queue)
        right=heapq.heappop(queue)
        parent=[left[0]+right[0],None,left,right]
        heapq.heappush(queue,parent)
    return queue[0]

def firstencode(Tree):
    codes={}
    def check(node,code):
        if node[1]:
            codes[node[1]]=code
        else:
            check(node[2],code+"0")
            check(node[3],code+"1")
    check(Tree,"")
    return codes

def encoding(string):
    encode=""
    for i in string:
        encode+=codes[i]
    return encode

def decoding(string):
    decode=""
    node=Tree
    for num in string:
        if num=="0":
            node=node[2]
        else:
            node=node[3]
        if node[1]:
            decode+=node[1]
            node=Tree
    return decode

n=int(input())
line=[]
for i in range(n):
    s,w=input().split()
    w=int(w)
    line.append((s,w))
Tree=tree(line)
codes=firstencode(Tree)
while True:
    try:
        string=input()
        if string[0] in ["0","1"]:
            print(decoding(string))
        else:
            print(encoding(string))
    except:
        break
```

代码运行截图 （AC代码截图，至少包含有"Accepted"）



#44392006提交状态

状态: Accepted

源代码

```
import heapq
def tree(line):
    queue=[]
    for s,w in line:
        heapq.heappush(queue,[w,s,None,None])
    while len(queue)>=2:
        left=heapq.heappop(queue)
        right=heapq.heappop(queue)
        parent=[left[0]+right[0],None,left,right]
        heapq.heappush(queue,parent)
    return queue[0]

def firstencode(Tree):
    codes={}
    def check(node,code):
        if node[1]:
            codes[node[1]]=code
        else:
            check(node[2],code+"0")
            check(node[3],code+"1")
    check(Tree,"")
    return codes

def encoding(string):
    encode=""
    for i in string:
        encode+=codes[i]
    return encode

def decoding(string):
    decode=""
    node=Tree
    for num in string:
        if num=="0":
```

基本信息

#: 44392006<br>
题目: 22161<br>
提交人: 23n2300011075(才疏学浅)<br>
内存: 3732kB<br>
时间: 23ms<br>
语言: Python3<br>
提交时间: 2024-03-25 00:01:12

## 晴问9.5: 平衡二叉树的建立

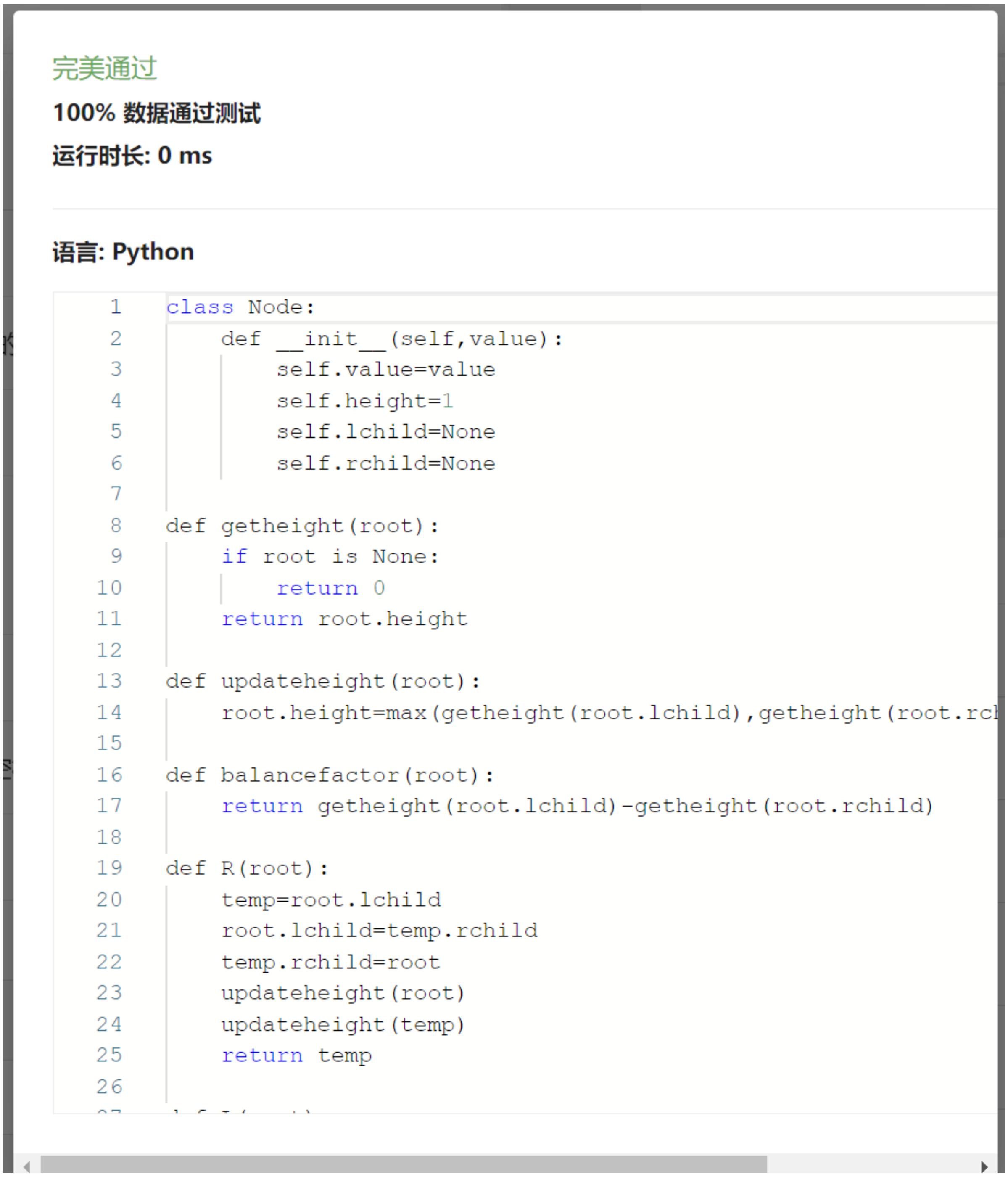
https://sunnywhy.com/sfbj/9/5/359

思路：AVL树写法

代码

```
#
class Node:
    def __init__(self,value):
        self.value=value
        self.height=1
        self.lchild=None
        self.rchild=None

def getheight(root):
    if root is None:
        return 0
    return root.height

def updateheight(root):
    root.height=max(getheight(root.lchild),getheight(root.rchild))+1

def balancefactor(root):
    return getheight(root.lchild)-getheight(root.rchild)

def R(root):
    temp=root.lchild
    root.lchild=temp.rchild
    temp.rchild=root
    updateheight(root)
    updateheight(temp)
    return temp

def L(root):
    temp=root.rchild
    root.rchild=temp.lchild
    temp.lchild=root
    updateheight(root)
    updateheight(temp)
    return temp

def insert(root,num):
    if root is None:
        return Node(num)
    if num<root.value:
        root.lchild=insert(root.lchild,num)
        updateheight(root)
        if balancefactor(root)==2:
            if balancefactor(root.lchild)==1:
                root=R(root)
            elif balancefactor(root.lchild)==-1:
                root.lchild=L(root.lchild)
                root=R(root)
    else:
        root.rchild=insert(root.rchild,num)
        updateheight(root)
        if balancefactor(root)==-2:
            if balancefactor(root.rchild)==-1:
                root=L(root)
            elif balancefactor(root.rchild)==1:
                root.rchild=R(root.rchild)
                root=L(root)
    return root

def preorder(root):
    ans=[]
    if root is None:
        return []
    ans.append(root.value)
    ans.extend(preorder(root.lchild))
    ans.extend(preorder(root.rchild))
    return ans

n=int(input())
l=list(map(int,input().split()))
tree=None
for num in l:
    tree=insert(tree,num)
print(*preorder(tree))
```

代码运行截图（AC代码截图，至少包含有"Accepted"）



完美通过

**100% 数据通过测试**

**运行时长: 0 ms**

**语言: Python**

```python
class Node:
    def __init__(self,value):
        self.value=value
        self.height=1
        self.lchild=None
        self.rchild=None

def getheight(root):
    if root is None:
        return 0
    return root.height

def updateheight(root):
    root.height=max(getheight(root.lchild),getheight(root.rc

def balancefactor(root):
    return getheight(root.lchild)-getheight(root.rchild)

def R(root):
    temp=root.lchild
    root.lchild=temp.rchild
    temp.rchild=root
    updateheight(root)
    updateheight(temp)
    return temp

def L(root):
```

## 02524: 宗教信仰

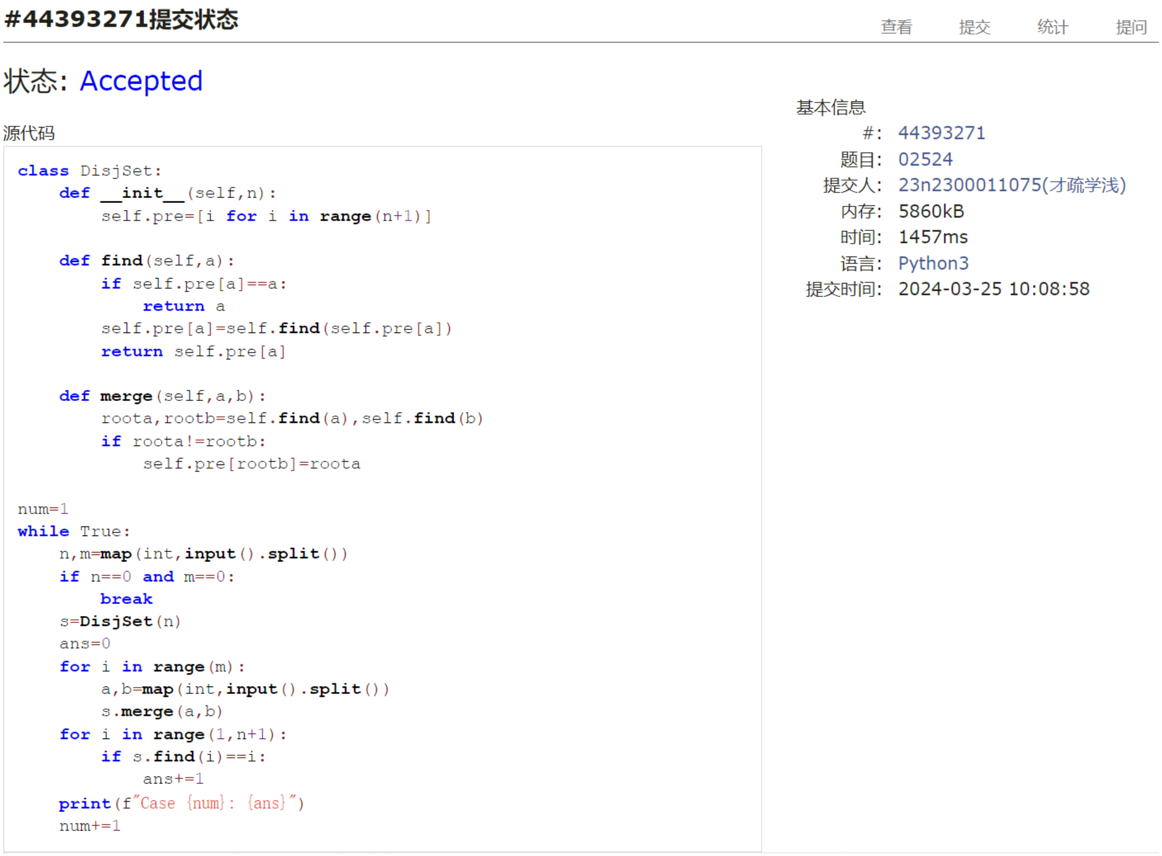
http://cs101.openjudge.cn/practice/02524/

思路：并查集

代码

```python
#
class DisjSet:
    def __init__(self,n):
        self.pre=[i for i in range(n+1)]

    def find(self,a):
        if self.pre[a]==a:
            return a
        self.pre[a]=self.find(self.pre[a])
        return self.pre[a]

    def merge(self,a,b):
        roota,rootb=self.find(a),self.find(b)
        if roota!=rootb:
            self.pre[rootb]=roota

num=1
while True:
    n,m=map(int,input().split())
    if n==0 and m==0:
        break
    s=DisjSet(n)
    ans=0
    for i in range(m):
        a,b=map(int,input().split())
        s.merge(a,b)
    for i in range(1,n+1):
        if s.find(i)==i:
            ans+=1
    print(f"Case {num}: {ans}")
    num+=1
```

代码运行截图 （AC代码截图，至少包含有"Accepted"）



#44393271提交状态

状态: Accepted

源代码

```python
class DisjSet:
    def __init__(self,n):
        self.pre=[i for i in range(n+1)]

    def find(self,a):
        if self.pre[a]==a:
            return a
        self.pre[a]=self.find(self.pre[a])
        return self.pre[a]

    def merge(self,a,b):
        roota,rootb=self.find(a),self.find(b)
        if roota!=rootb:
            self.pre[rootb]=roota

num=1
while True:
    n,m=map(int,input().split())
    if n==0 and m==0:
        break
    s=DisjSet(n)
    ans=0
    for i in range(m):
        a,b=map(int,input().split())
        s.merge(a,b)
    for i in range(1,n+1):
        if s.find(i)==i:
            ans+=1
    print(f"Case {num}: {ans}")
    num+=1
```

基本信息

#: 44393271
题目: 02524
提交人: 23n2300011075(才疏学浅)
内存: 5860kB
时间: 1457ms
语言: Python3
提交时间: 2024-03-25 10:08:58

## 2. 学习总结和收获

如果作业题目简单，有否额外练习题目，比如：OJ“2024spring每日选做”、CF、LeetCode、洛谷等网站题目。

这几次作业都在使用树的写法，正在尝试将所有函数及树节点全部在新定义的树类数据结构下书写调用

并查集来了，线段树还会远吗？