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

Updated 2214 GMT+8 March 24, 2024

2024 spring, Complied by 夏天明 元培学院

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

操作系统：Windows 10 | 22H2

Python编程环境：Spyder IDE 5.4.3 | Python 3.11.4 64-bit

1. 题目

22275: 二叉搜索树的遍历

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

思路：用二叉搜索树节点的大小关系性质判断两颗子树的位置，然后递归

代码

```python
def getPost(pre):
    if not pre:
        return []
    i = 0
    while i < len(pre) and pre[i] <= pre[0]:
        i += 1
    return getPost(pre[1:i]) + getPost(pre[i:]) + [pre[0]]

n = int(input())
pre = [int(i) for i in input().split()]
print(*getPost(pre))
```

代码运行截图

05455: 二叉搜索树的层次遍历

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

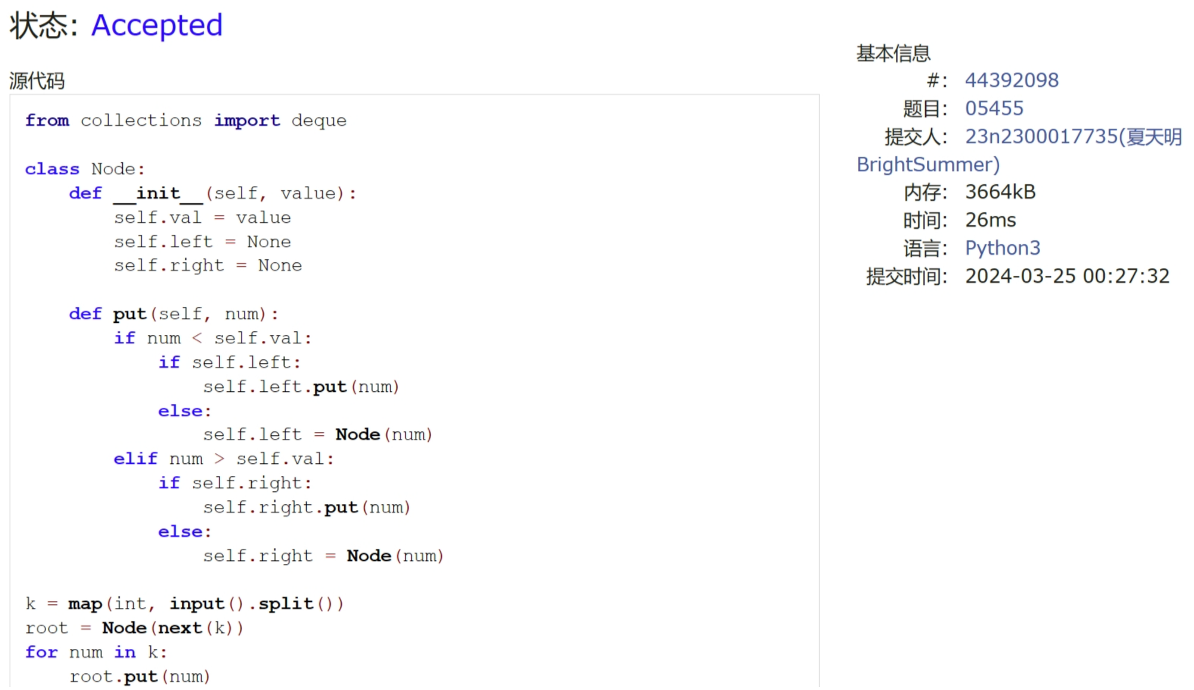
思路：直接建树

代码

```python
from collections import deque

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

    def put(self, num):
        if num < self.val:
            if self.left:
                self.left.put(num)
            else:
                self.left = Node(num)
        elif num > self.val:
            if self.right:
                self.right.put(num)
            else:
                self.right = Node(num)

k = map(int, input().split())
```

```
root = Node(next(k))
for num in k:
    root.put(num)
q = deque([root])
ans = []
while q:
    nd = q.popleft()
    ans.append(nd.val)
    if nd.left:
        q.append(nd.left)
    if nd.right:
        q.append(nd.right)
print(*ans)
```

代码运行截图



状态: Accepted

源代码

```
from collections import deque

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

    def put(self, num):
        if num < self.val:
            if self.left:
                self.left.put(num)
            else:
                self.left = Node(num)
        elif num > self.val:
            if self.right:
                self.right.put(num)
            else:
                self.right = Node(num)

k = map(int, input().split())
root = Node(next(k))
for num in k:
    root.put(num)
```

基本信息

#:	44392098
题目:	05455
提交人:	23n2300017735(夏天明 BrightSummer)
内存:	3664kB
时间:	26ms
语言:	Python3
提交时间:	2024-03-25 00:27:32

04078: 实现堆结构

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

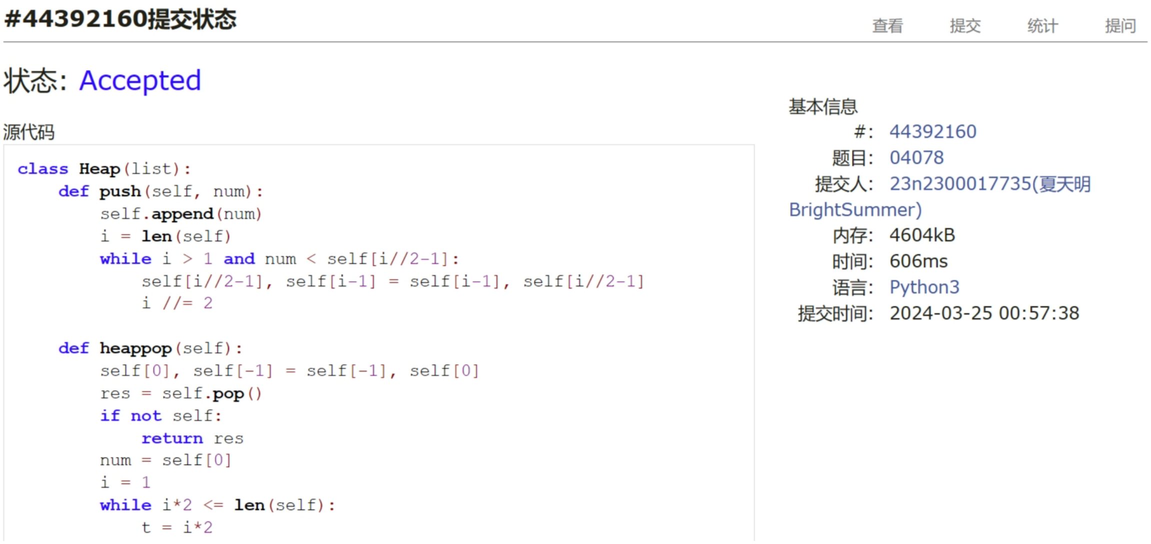
练习自己写个BinHeap。当然机考时候，如果遇到这样题目，直接import heapq。手搓栈、队列、堆、AVL等，考试前需要搓个遍。

思路：直接实现。继承list类，按照完全二叉树的非嵌套列表法表示，更方便

代码

```python
class Heap(list):
    def push(self, num):
        self.append(num)
        i = len(self)
        while i > 1 and num < self[i//2-1]:
            self[i//2-1], self[i-1] = self[i-1], self[i//2-1]
            i //= 2

    def heappop(self):
        self[0], self[-1] = self[-1], self[0]
        res = self.pop()
        if not self:
            return res
        num = self[0]
        i = 1
        while i*2 <= len(self):
            t = i*2
            if t < len(self) and self[t] < self[t-1]:
                t += 1
            if num <= self[t-1]:
                break
            self[i-1], self[t-1] = self[t-1], self[i-1]
            i = t
        return res

heap = Heap()
for o in range(int(input())):
    s = [int(i) for i in input().split()]
    if s[0] == 1:
        heap.push(s[1])
    else:
        print(heap.heappop())
```

代码运行截图



#44392160提交状态

状态: Accepted

源代码

```python
class Heap(list):
    def push(self, num):
        self.append(num)
        i = len(self)
        while i > 1 and num < self[i//2-1]:
            self[i//2-1], self[i-1] = self[i-1], self[i//2-1]
            i //= 2

    def heappop(self):
        self[0], self[-1] = self[-1], self[0]
        res = self.pop()
        if not self:
            return res
        num = self[0]
        i = 1
        while i*2 <= len(self):
            t = i*2
```

基本信息

#: 44392160

题目: 04078

提交人: 23n2300017735(夏天明BrightSummer)

内存: 4604kB

时间: 606ms

语言: Python3

提交时间: 2024-03-25 00:57:38

22161: 哈夫曼编码树

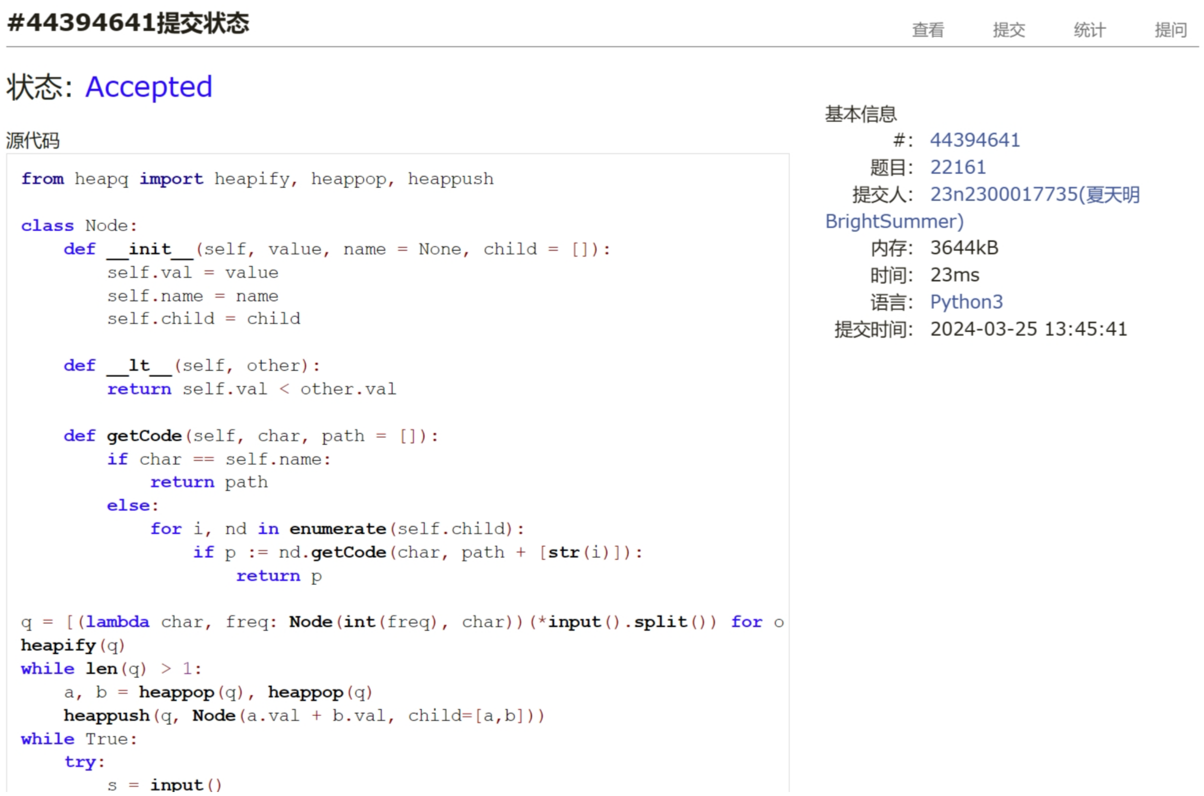
http://cs101.openjudge.cn/practice/22161/

思路：直接实现

代码

```python
from heapq import heapify, heappop, heappush

class Node:
    def __init__(self, value, name, child = []):
        self.val = value
        self.name = name
        self.child = child

    def __lt__(self, other):
        return (self.val, self.name) < (other.val, other.name)

    def getCode(self, char, path = []):
        if char == self.name:
            return path
        else:
            for i, nd in enumerate(self.child):
                if p := nd.getCode(char, path + [str(i)]):
                    return p

q = [(lambda char, freq: Node(int(freq), char))(*input().split()) for o in range(int(input()))]
heapify(q)
while len(q) > 1:
    a, b = heappop(q), heappop(q)
    heappush(q, Node(a.val + b.val, a.name + b.name, child=[a,b]))
while True:
    try:
        s = input()
    except EOFError:
        break
    if s.isdigit():
        ans = []
        curr = q[0]
        for code in map(int, s + '0'):
            if not curr.child:
                ans.append(curr.name)
                curr = q[0]
            curr = curr.child[code]
        print(''.join(ans))
    else:
        print(''.join(code for char in s for code in q[0].getCode(char)))
```

代码运行截图



#44394641提交状态

状态: Accepted

源代码

```
from heapq import heapify, heappop, heappush

class Node:
    def __init__(self, value, name = None, child = []):
        self.val = value
        self.name = name
        self.child = child

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

    def getCode(self, char, path = []):
        if char == self.name:
            return path
        else:
            for i, nd in enumerate(self.child):
                if p := nd.getCode(char, path + [str(i)]):
                    return p

q = [(lambda char, freq: Node(int(freq), char))(*input().split()) for o
heapify(q)
while len(q) > 1:
    a, b = heappop(q), heappop(q)
    heappush(q, Node(a.val + b.val, child=[a,b]))
while True:
    try:
        s = input()
```

基本信息

#:	44394641
题目:	22161
提交人:	23n2300017735(夏天明 BrightSummer)
内存:	3644kB
时间:	23ms
语言:	Python3
提交时间:	2024-03-25 13:45:41

晴问9.5: 平衡二叉树的建立

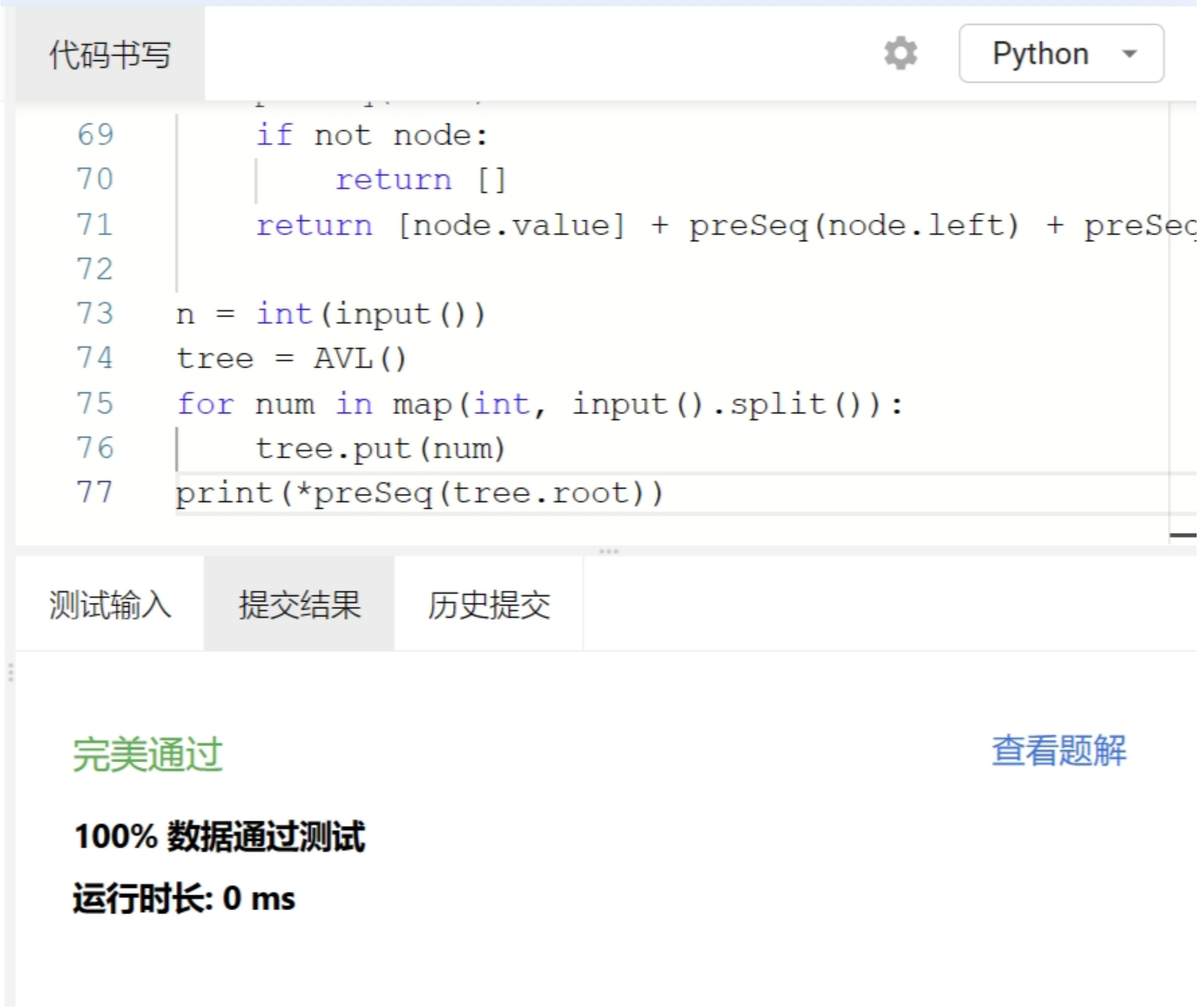
https://sunnywhy.com/sfbj/9/5/359

思路：直接实现，但是比较难写。一开始参考谢老师ppt的写法，比较复杂，出现各种错误，没能AC。后来重写了老师课件里的代码，AC了。

代码

```
class Node:
    def __init__(self, value):
        self.value = value
        self.left = None
        self.right = None
        self.height = 1

class AVL:
    def __init__(self):
        self.root = None
    def put(self, value):
        self.root = self._put(value, self.root)

    def _put(self, value, node):
        if not node:
            return Node(value)
        if value < node.value:
            node.left = self._put(value, node.left)
        else:
            node.right = self._put(value, node.right)
        self._update_height(node)
        if (balance:=self._get_balance(node)) > 1:
            if value < node.left.value: # 树形是 LL
                return self._rotate_right(node)
            else:   # 树形是 LR
                node.left = self._rotate_left(node.left)
                return self._rotate_right(node)
        if balance < -1:
            if value > node.right.value:    # 树形是 RR
                return self._rotate_left(node)
            else:   # 树形是 RL
                node.right = self._rotate_right(node.right)
                return self._rotate_left(node)
        return node

    def _get_height(self, node):
        if not node:
            return 0
        return node.height

    def _update_height(self, node):
        node.height = 1 + max(self._get_height(node.left),
self._get_height(node.right))

    def _get_balance(self, node):
        if not node:
            return 0
        return self._get_height(node.left) - self._get_height(node.right)

    def _rotate_left(self, z):
        y = z.right
        T2 = y.left
        y.left = z
        z.right = T2
        self._update_height(z)
        self._update_height(y)
        return y

    def _rotate_right(self, y):
        x = y.left
        T2 = x.right
        x.right = y
        y.left = T2
        self._update_height(y)
        self._update_height(x)
        return x

def preSeq(node):
    if not node:
        return []
    return [node.value] + preSeq(node.left) + preSeq(node.right)

n = int(input())
tree = AVL()
for num in map(int, input().split()):
    tree.put(num)
print(*preSeq(tree.root))
```

代码运行截图



```
    if not node:
        return []
    return [node.value] + preSeq(node.left) + preSeq

n = int(input())
tree = AVL()
for num in map(int, input().split()):
    tree.put(num)
print(*preSeq(tree.root))
```

完美通过

100% 数据通过测试

运行时长: 0 ms

02524: 宗教信仰

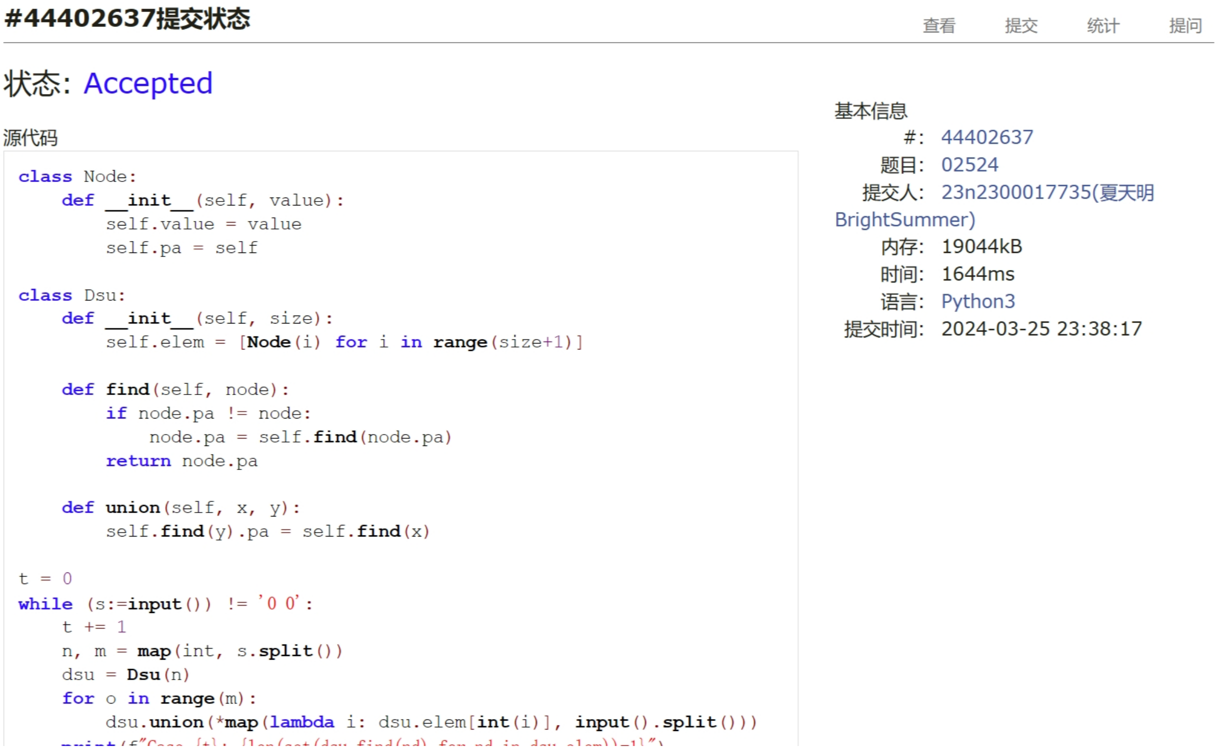
http://cs101.openjudge.cn/practice/02524/

思路：使用并查集直接实现

代码

```python
class Node:
    def __init__(self, value):
        self.value = value
        self.pa = self

class Dsu:
    def __init__(self, size):
        self.elem = [Node(i) for i in range(size+1)]

    def find(self, node):
        if node.pa != node:
            node.pa = self.find(node.pa)
        return node.pa

    def union(self, x, y):
        self.find(y).pa = self.find(x)

t = 0
while (s:=input()) != '0 0':
    t += 1
    n, m = map(int, s.split())
    dsu = Dsu(n)
    for o in range(m):
        dsu.union(*map(lambda i: dsu.elem[int(i)], input().split()))
    print(f"Case {t}: {len(set(dsu.find(nd) for nd in dsu.elem))-1}")
```

代码运行截图



#44402637提交状态

状态: Accepted

源代码

```
class Node:
    def __init__(self, value):
        self.value = value
        self.pa = self

class Dsu:
    def __init__(self, size):
        self.elem = [Node(i) for i in range(size+1)]

    def find(self, node):
        if node.pa != node:
            node.pa = self.find(node.pa)
        return node.pa

    def union(self, x, y):
        self.find(y).pa = self.find(x)

t = 0
while (s:=input()) != '0 0':
    t += 1
    n, m = map(int, s.split())
    dsu = Dsu(n)
    for o in range(m):
        dsu.union(*map(lambda i: dsu.elem[int(i)], input().split()))
    print(f"Case {t}: {len(set(dsu.find(nd) for nd in dsu.elem))-1}")
```

基本信息

#: 44402637
题目: 02524
提交人: 23n2300017735(夏天明BrightSummer)
内存: 19044kB
时间: 1644ms
语言: Python3
提交时间: 2024-03-25 23:38:17

2. 学习总结和收获

本次作业涉及比较高级的数据结构，进一步熟悉类的使用，加深对树的结构和递归的理解