Assignment #8: 图论：概念、遍历，及 树算

Updated 1919 GMT+8 Apr 8, 2024

2024 spring, Complied by 张羽扬 数学科学学院

说明：

1）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。

2）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。

3）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 11 家庭中文版 22H2

Python编程环境：Spyder IDE 5.2.2

C/C++编程环境：无

1. 题目

19943: 图的拉普拉斯矩阵

matrices, http://cs101.openjudge.cn/practice/19943/

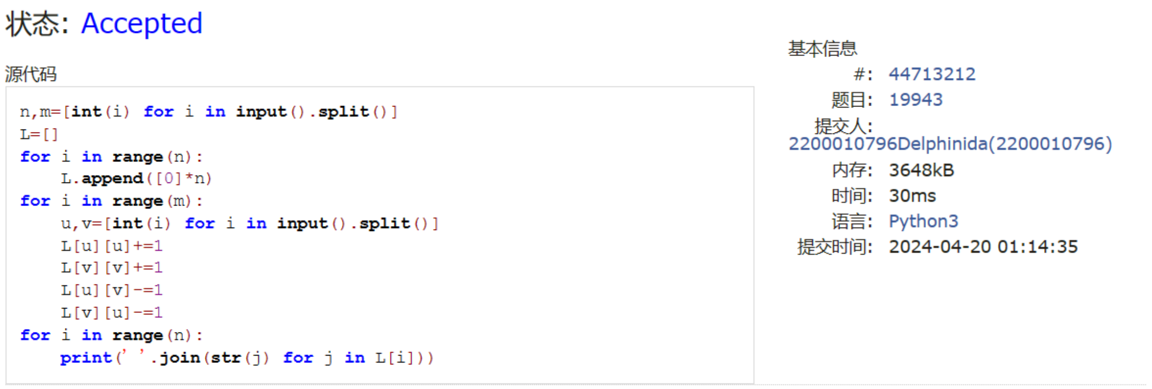
请定义Vertex类，Graph类，然后实现

思路：

代码

```
n,m=[int(i) for i in input().split()]
L=[]
for i in range(n):
    L.append([0]*n)
for i in range(m):
    u,v=[int(i) for i in input().split()]
    L[u][u]+=1
    L[v][v]+=1
    L[u][v]-=1
    L[v][u]-=1
for i in range(n):
    print(' '.join(str(j) for j in L[i]))
```

代码运行截图



状态: Accepted

源代码

```
n,m=[int(i) for i in input().split()]
L=[]
for i in range(n):
    L.append([0]*n)
for i in range(m):
    u,v=[int(i) for i in input().split()]
    L[u][u]+=1
    L[v][v]+=1
    L[u][v]-=1
    L[v][u]-=1
for i in range(n):
    print(' '.join(str(j) for j in L[i]))
```

基本信息

#: 44713212

题目: 19943

提交人: 2200010796Delphinida(2200010796)

内存: 3648kB

时间: 30ms

语言: Python3

提交时间: 2024-04-20 01:14:35

18160: 最大连通域面积

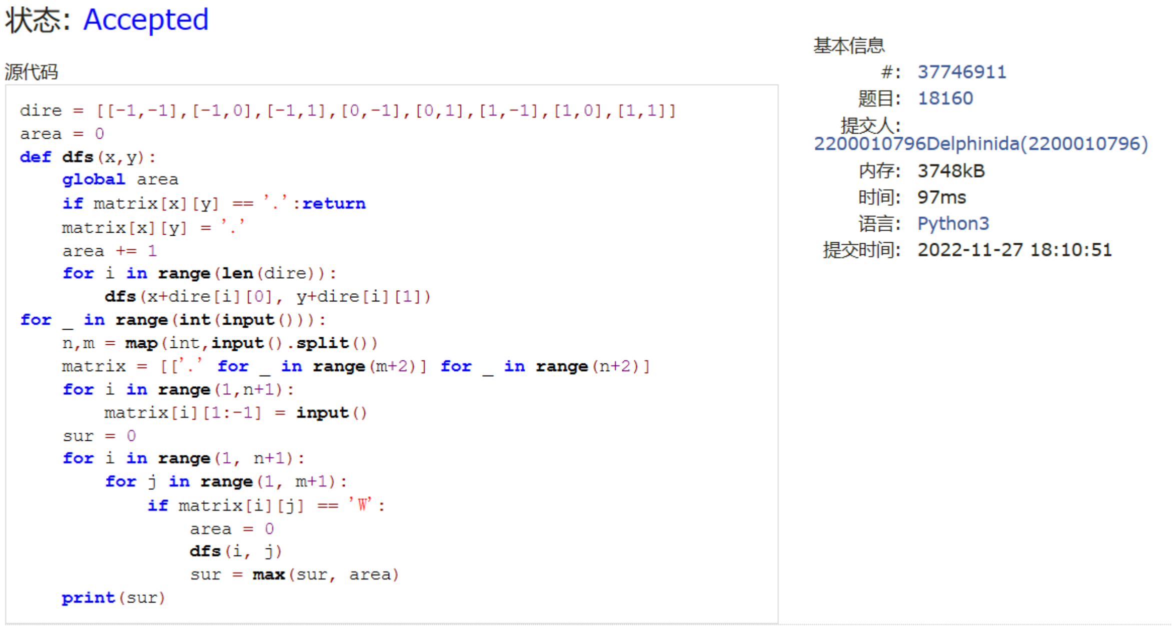
matrix/dfs similar, http://cs101.openjudge.cn/practice/18160

思路:

代码

```
dire = [[-1,-1],[-1,0],[-1,1],[0,-1],[0,1],[1,-1],[1,0],[1,1]]
area = 0
def dfs(x,y):
    global area
    if matrix[x][y] == '.':return
    matrix[x][y] = '.'
    area += 1
    for i in range(len(dire)):
        dfs(x+dire[i][0], y+dire[i][1])
for _ in range(int(input())):
    n,m = map(int,input().split())
    matrix = [['.' for _ in range(m+2)] for _ in range(n+2)]
    for i in range(1,n+1):
        matrix[i][1:-1] = input()
    sur = 0
    for i in range(1, n+1):
        for j in range(1, m+1):
            if matrix[i][j] == 'W':
                area = 0
                dfs(i, j)
                sur = max(sur, area)
    print(sur)
```

代码运行截图



状态: Accepted

源代码

```
dire = [[-1,-1],[-1,0],[-1,1],[0,-1],[0,1],[1,-1],[1,0],[1,1]]
area = 0
def dfs(x,y):
    global area
    if matrix[x][y] == '.':return
    matrix[x][y] = '.'
    area += 1
    for i in range(len(dire)):
        dfs(x+dire[i][0], y+dire[i][1])
for _ in range(int(input())):
    n,m = map(int,input().split())
    matrix = [['.' for _ in range(m+2)] for _ in range(n+2)]
    for i in range(1,n+1):
        matrix[i][1:-1] = input()
    sur = 0
    for i in range(1, n+1):
        for j in range(1, m+1):
            if matrix[i][j] == 'W':
                area = 0
                dfs(i, j)
                sur = max(sur, area)
    print(sur)
```

基本信息

#: 37746911
题目: 18160
提交人: 2200010796Delphinida(2200010796)
内存: 3748kB
时间: 97ms
语言: Python3
提交时间: 2022-11-27 18:10:51

sy383: 最大权值连通块

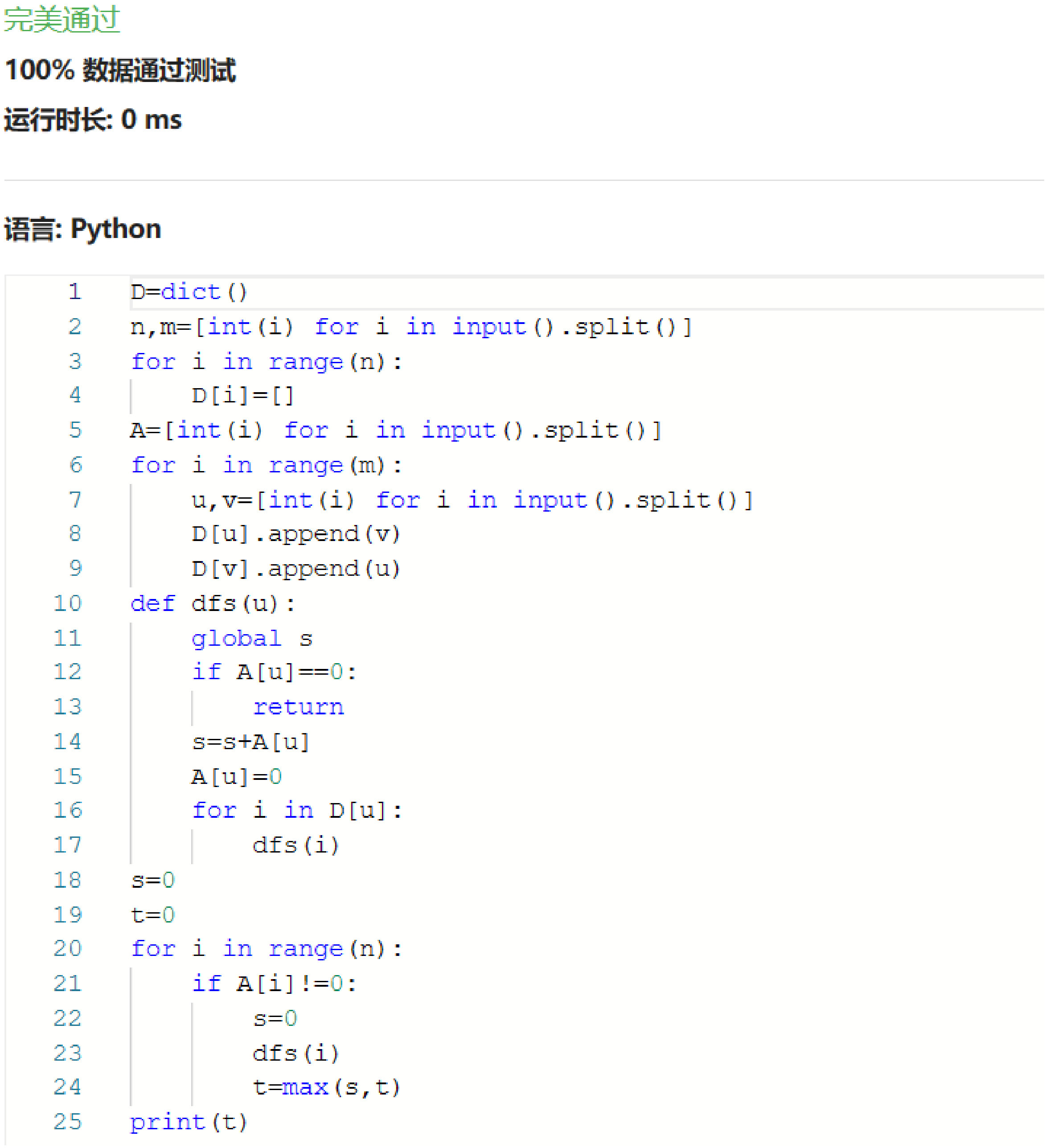
https://sunnywhy.com/sfbj/10/3/383

思路：和上题类似 只是二维的图形变成了图

代码

```
D=dict()
n,m=[int(i) for i in input().split()]
for i in range(n):
    D[i]=[]
A=[int(i) for i in input().split()]
for i in range(m):
    u,v=[int(i) for i in input().split()]
    D[u].append(v)
    D[v].append(u)
def dfs(u):
    global s
    if A[u]==0:
        return
    s=s+A[u]
    A[u]=0
    for i in D[u]:
        dfs(i)
s=0
t=0
for i in range(n):
    if A[i]!=0:
        s=0
        dfs(i)
        t=max(s,t)
print(t)
```

代码运行截图



完美通过

100% 数据通过测试

运行时长: 0 ms

语言: Python

```
D=dict()
n,m=[int(i) for i in input().split()]
for i in range(n):
    D[i]=[]
A=[int(i) for i in input().split()]
for i in range(m):
    u,v=[int(i) for i in input().split()]
    D[u].append(v)
    D[v].append(u)
def dfs(u):
    global s
    if A[u]==0:
        return
    s=s+A[u]
    A[u]=0
    for i in D[u]:
        dfs(i)
s=0
t=0
for i in range(n):
    if A[i]!=0:
        s=0
        dfs(i)
        t=max(s,t)
print(t)
```

03441: 4 Values whose Sum is 0

data structure/binary search, http://cs101.openjudge.cn/practice/03441

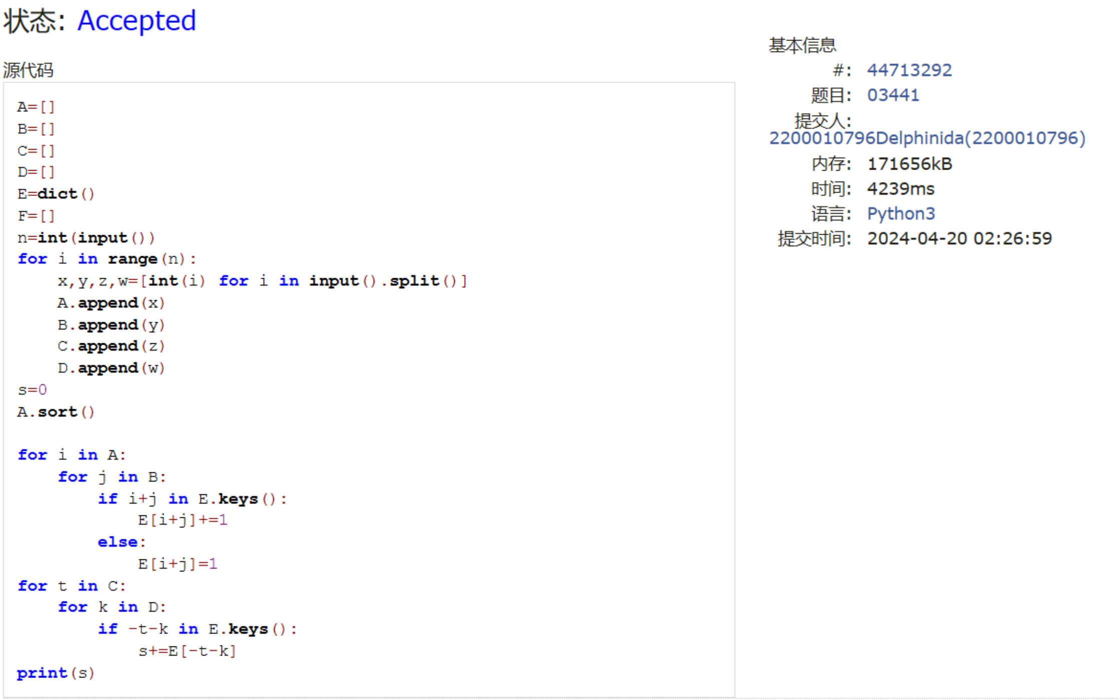
思路:

代码

```
A=[]
B=[]
C=[]
D=[]
E=dict()
F=[]
n=int(input())
for i in range(n):
    x,y,z,w=[int(i) for i in input().split()]
    A.append(x)
    B.append(y)
    C.append(z)
    D.append(w)
s=0
A.sort()

for i in A:
    for j in B:
        if i+j in E.keys():
            E[i+j]+=1
        else:
            E[i+j]=1
for t in C:
    for k in D:
        if -t-k in E.keys():
            s+=E[-t-k]
print(s)
```

代码运行截图



状态: Accepted

源代码

```
A=[]
B=[]
C=[]
D=[]
E=dict()
F=[]
n=int(input())
for i in range(n):
    x,y,z,w=[int(i) for i in input().split()]
    A.append(x)
    B.append(y)
    C.append(z)
    D.append(w)
s=0
A.sort()

for i in A:
    for j in B:
        if i+j in E.keys():
            E[i+j]+=1
        else:
            E[i+j]=1
for t in C:
    for k in D:
        if -t-k in E.keys():
            s+=E[-t-k]
print(s)
```

基本信息

#: 44713292
题目: 03441
提交人: 2200010796Delphinida(2200010796)
内存: 171656kB
时间: 4239ms
语言: Python3
提交时间: 2024-04-20 02:26:59

04089: 电话号码

trie, http://cs101.openjudge.cn/practice/04089/

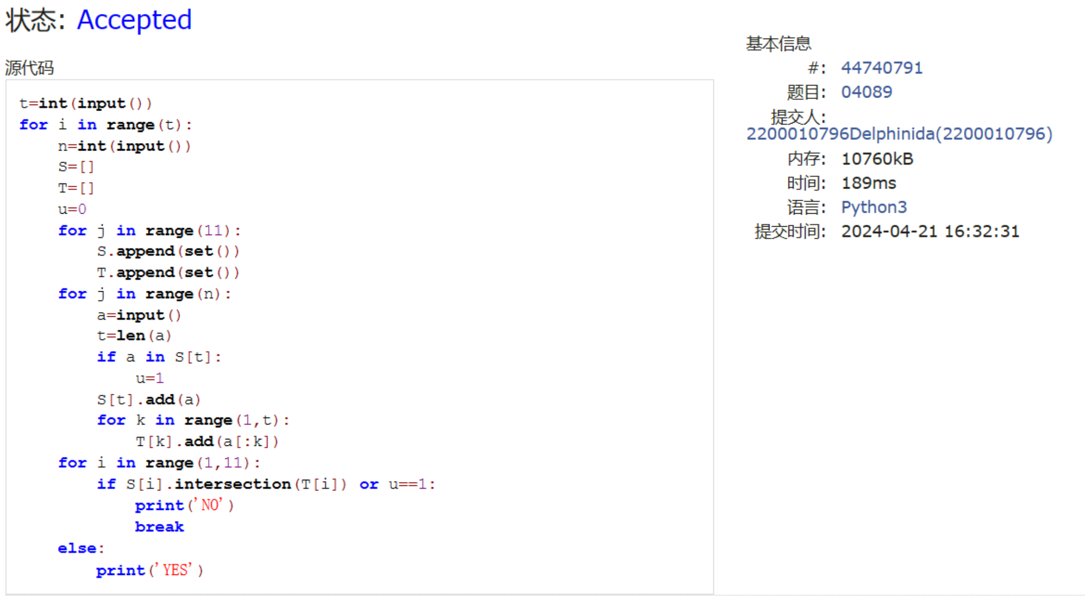
Trie 数据结构可能需要自学下。

思路：好像没有使用trie数据结构？ 只是把每个输入的前缀序列全都储存了下来 因为电话的长度只有10所以好像没问题

代码

```python
t=int(input())
for i in range(t):
    n=int(input())
    S=[]
    T=[]
    u=0
    for j in range(11):
        S.append(set())
        T.append(set())
    for j in range(n):
        a=input()
        t=len(a)
        if a in S[t]:
            u=1
        S[t].add(a)
        for k in range(1,t):
            T[k].add(a[:k])
    for i in range(1,11):
        if S[i].intersection(T[i]) or u==1:
            print('NO')
            break
    else:
        print('YES')
```

代码运行截图



状态: Accepted

源代码

```
t=int(input())
for i in range(t):
    n=int(input())
    S=[]
    T=[]
    u=0
    for j in range(11):
        S.append(set())
        T.append(set())
    for j in range(n):
        a=input()
        t=len(a)
        if a in S[t]:
            u=1
        S[t].add(a)
        for k in range(1,t):
            T[k].add(a[:k])
    for i in range(1,11):
        if S[i].intersection(T[i]) or u==1:
            print('NO')
            break
    else:
        print('YES')
```

基本信息

#: 44740791

题目: 04089

提交人: 2200010796Delphinida(2200010796)

内存: 10760kB

时间: 189ms

语言: Python3

提交时间: 2024-04-21 16:32:31

04082: 树的镜面映射

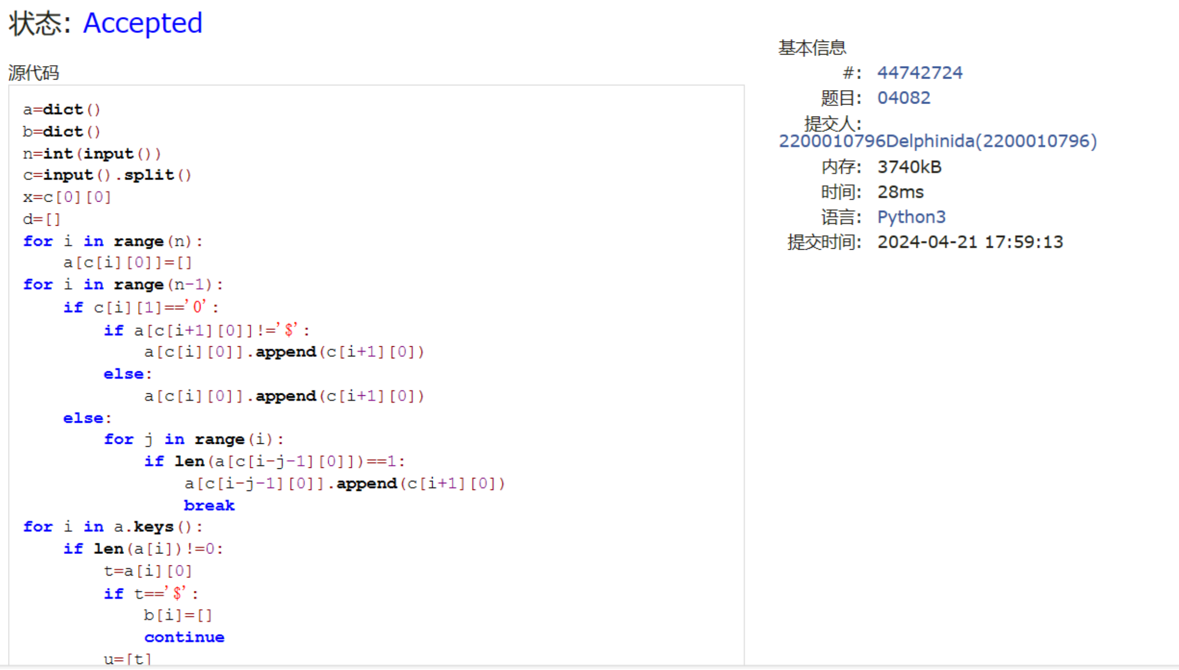
http://cs101.openjudge.cn/practice/04082/

思路:

代码

```
a=dict()
b=dict()
n=int(input())
c=input().split()
x=c[0][0]
d=[]
for i in range(n):
    a[c[i][0]]=[]
for i in range(n-1):
    if c[i][1]=='0':
        if a[c[i+1][0]]!='$':
            a[c[i][0]].append(c[i+1][0])
        else:
            a[c[i][0]].append(c[i+1][0])
    else:
        for j in range(i):
            if len(a[c[i-j-1][0]])==1:
                a[c[i-j-1][0]].append(c[i+1][0])
                break
for i in a.keys():
    if len(a[i])!=0:
        t=a[i][0]
        if t=='$':
            b[i]=[]
            continue
        u=[t]
        while len(a[t])!=0:
            t=a[t][1]
            if t!='$':
                u.append(t)
        u.reverse()
        b[i]=u
    elif i!='$':
        b[i]=[]
m=[x]
while m!=[]:
    d+=m
    p=[]
    for i in m:
        p+=b[i]
    m=p
print(' '.join(d))
```

代码运行截图



状态: Accepted

源代码

```
a=dict()
b=dict()
n=int(input())
c=input().split()
x=c[0][0]
d=[]
for i in range(n):
    a[c[i][0]]=[]
for i in range(n-1):
    if c[i][1]==’0’:
        if a[c[i+1][0]]!=’$’:
            a[c[i][0]].append(c[i+1][0])
        else:
            a[c[i][0]].append(c[i+1][0])
    else:
        for j in range(i):
            if len(a[c[i-j-1][0]])==1:
                a[c[i-j-1][0]].append(c[i+1][0])
                break
for i in a.keys():
    if len(a[i])!=0:
        t=a[i][0]
        if t==’$’:
            b[i]=[]
            continue
        u=[t]
```

基本信息

#: 44742724

题目: 04082

提交人: 2200010796Delphinida(2200010796)

内存: 3740kB

时间: 28ms

语言: Python3

提交时间: 2024-04-21 17:59:13

2. 学习总结和收获

最后一题花了很久，树的各种形式的转化还是需要熟练