Assignment #8: 图论：概念、遍历，及 树算

Updated 1919 GMT+8 Apr 8, 2024

2024 spring, Complied by 同学的姓名、院系

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

（请改为同学的操作系统、编程环境等）

操作系统：macOS Ventura 13.4.1 (c)

Python编程环境：Spyder IDE 5.2.2, PyCharm 2023.1.4 (Professional Edition)

C/C++编程环境：Mac terminal vi (version 9.0.1424), g++/gcc (Apple clang version 14.0.3, clang-1403.0.22.14.1)

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

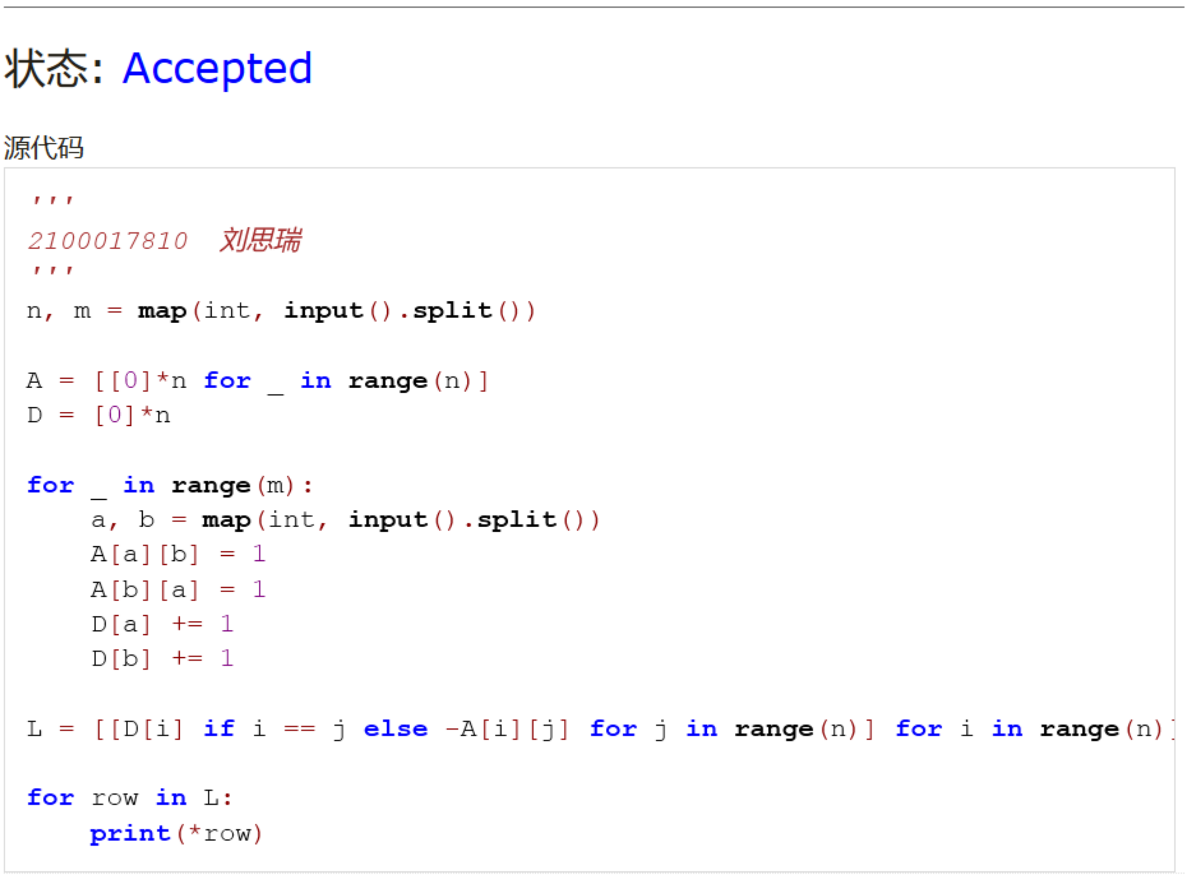
思路：

正常的操作问题

代码

```python
'''
2100017810   刘思瑞
'''
n, m = map(int, input().split())

A = [[0]*n for _ in range(n)]
D = [0]*n

for _ in range(m):
    a, b = map(int, input().split())
    A[a][b] = 1
    A[b][a] = 1
    D[a] += 1
    D[b] += 1

L = [[D[i] if i == j else -A[i][j] for j in range(n)] for i in range(n)]

for row in L:
    print(*row)
```

代码运行截图



状态: Accepted

源代码

```python
'''
2100017810   刘思瑞
'''
n, m = map(int, input().split())

A = [[0]*n for _ in range(n)]
D = [0]*n

for _ in range(m):
    a, b = map(int, input().split())
    A[a][b] = 1
    A[b][a] = 1
    D[a] += 1
    D[b] += 1

L = [[D[i] if i == j else -A[i][j] for j in range(n)] for i in range(n)]

for row in L:
    print(*row)
```

18160: 最大连通域面积

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

思路：

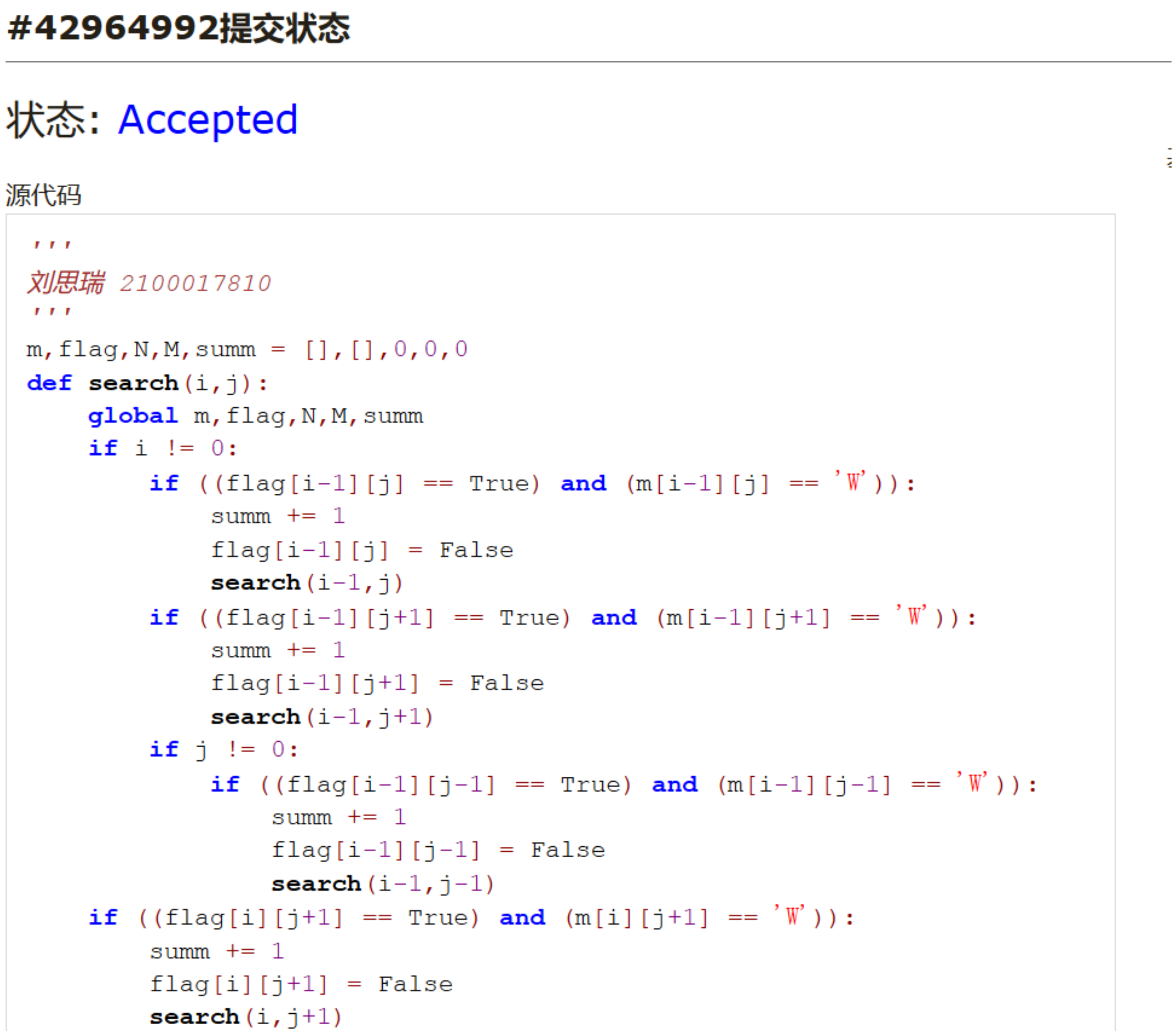
dfs

代码

```python
'''
刘思瑞 2100017810
```

```python
'''
m,flag,N,M,summ = [],[],0,0,0
def search(i,j):
    global m,flag,N,M,summ
    if i != 0:
        if ((flag[i-1][j] == True) and (m[i-1][j] == 'W')):
            summ += 1
            flag[i-1][j] = False
            search(i-1,j)
        if ((flag[i-1][j+1] == True) and (m[i-1][j+1] == 'W')):
            summ += 1
            flag[i-1][j+1] = False
            search(i-1,j+1)
        if j != 0:
            if ((flag[i-1][j-1] == True) and (m[i-1][j-1] == 'W')):
                summ += 1
                flag[i-1][j-1] = False
                search(i-1,j-1)
    if ((flag[i][j+1] == True) and (m[i][j+1] == 'W')):
        summ += 1
        flag[i][j+1] = False
        search(i,j+1)
    if ((flag[i+1][j+1] == True) and (m[i+1][j+1] == 'W')):
        summ += 1
        flag[i+1][j+1] = False
        search(i+1,j+1)
    if ((flag[i+1][j] == True) and (m[i+1][j] == 'W')):
        summ += 1
        flag[i+1][j] = False
        search(i+1,j)
    if j != 0:
        if ((flag[i][j-1] == True) and (m[i][j-1] == 'W')):
            summ += 1
            flag[i][j-1] = False
            search(i,j-1)
        if ((flag[i+1][j-1] == True) and (m[i+1][j-1] == 'W')):
            summ += 1
            flag[i+1][j-1] = False
            search(i+1,j-1)
    return


num = int(input())
for k in range(num):
    m = []
    flag = []
    sum = 0
    N,M = map(int,input().split())
    for i in range(N):
        flag.append([True]*(M)+[False])
        s = input()
        temp = []
        for j in range(M):
            temp.append(s[j])
        temp.append('.')
        m.append(temp)
    m.append(['.']*(M+1))
    flag.append([False]*(M+1))
    for i in range(N):
        for j in range(M):
            if m[i][j] =='W' and flag[i][j] == True:
                summ = 1
                flag[i][j] = False
                search(i,j)
                sum = max(sum,summ)
    print(sum)
```

代码运行截图



#42964992提交状态

状态: Accepted

源代码

```
'''
刘思瑞 2100017810
'''
m,flag,N,M,summ = [],[],0,0,0
def search(i,j):
    global m,flag,N,M,summ
    if i != 0:
        if ((flag[i-1][j] == True) and (m[i-1][j] == 'W')):
            summ += 1
            flag[i-1][j] = False
            search(i-1,j)
        if ((flag[i-1][j+1] == True) and (m[i-1][j+1] == 'W')):
            summ += 1
            flag[i-1][j+1] = False
            search(i-1,j+1)
        if j != 0:
            if ((flag[i-1][j-1] == True) and (m[i-1][j-1] == 'W')):
                summ += 1
                flag[i-1][j-1] = False
                search(i-1,j-1)
    if ((flag[i][j+1] == True) and (m[i][j+1] == 'W')):
        summ += 1
        flag[i][j+1] = False
        search(i,j+1)
```

sy383: 最大权值连通块

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

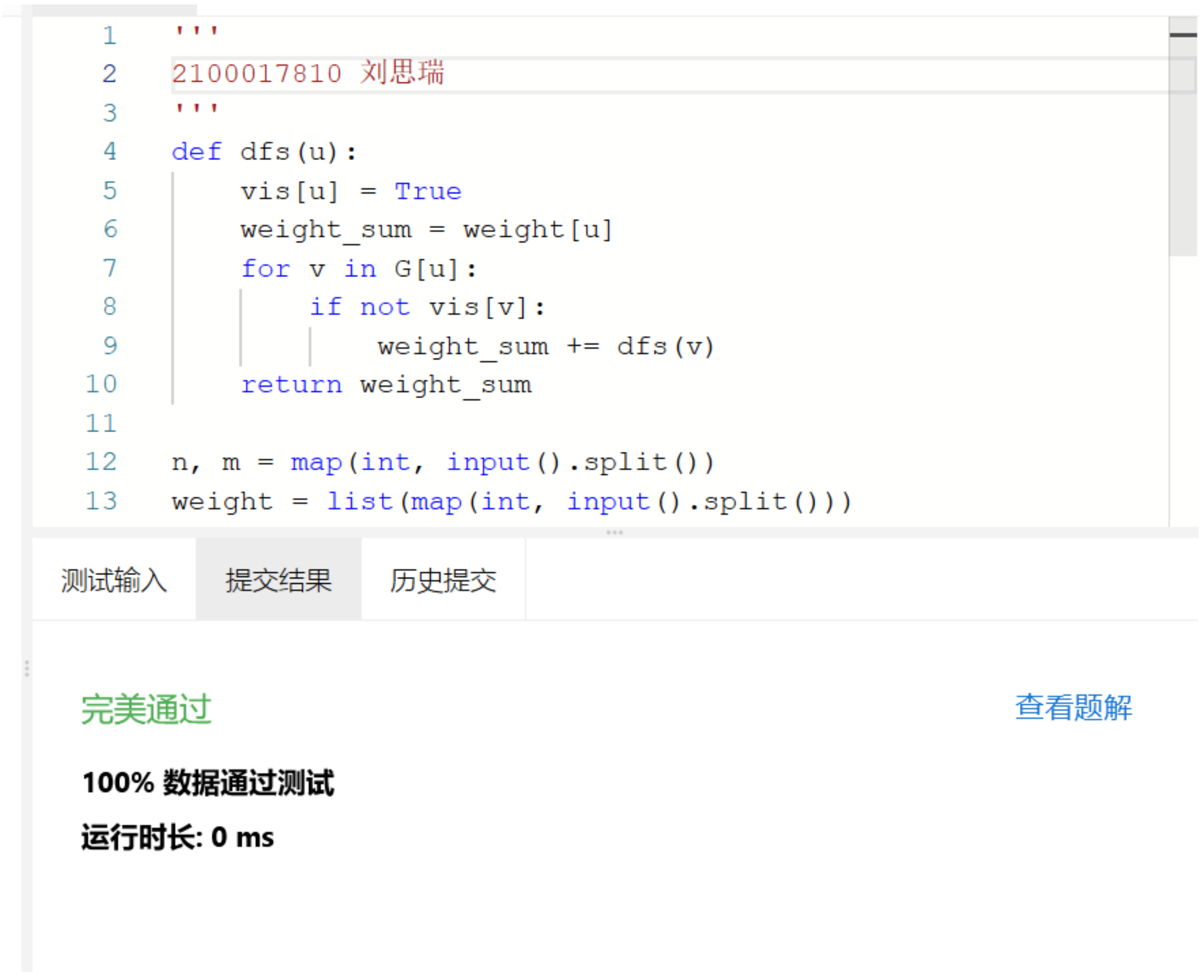
思路:

dfs

代码

```python
'''
2100017810 刘思瑞
'''
def dfs(u):
    vis[u] = True
    weight_sum = weight[u]
    for v in G[u]:
        if not vis[v]:
            weight_sum += dfs(v)
    return weight_sum

n, m = map(int, input().split())
weight = list(map(int, input().split()))
G = [[] for _ in range(n)]
vis = [False] * n

for _ in range(m):
    u, v = map(int, input().split())
    G[u].append(v)
    G[v].append(u)

max_weight_sum = 0
for i in range(n):
    if not vis[i]:
        max_weight_sum = max(max_weight_sum, dfs(i))

print(max_weight_sum)
```

代码运行截图



```
'''
2100017810 刘思瑞
'''
def dfs(u):
    vis[u] = True
    weight_sum = weight[u]
    for v in G[u]:
        if not vis[v]:
            weight_sum += dfs(v)
    return weight_sum

n, m = map(int, input().split())
weight = list(map(int, input().split()))
```

完美通过

查看题解

100% 数据通过测试

运行时长: 0 ms

03441: 4 Values whose Sum is 0

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

思路:

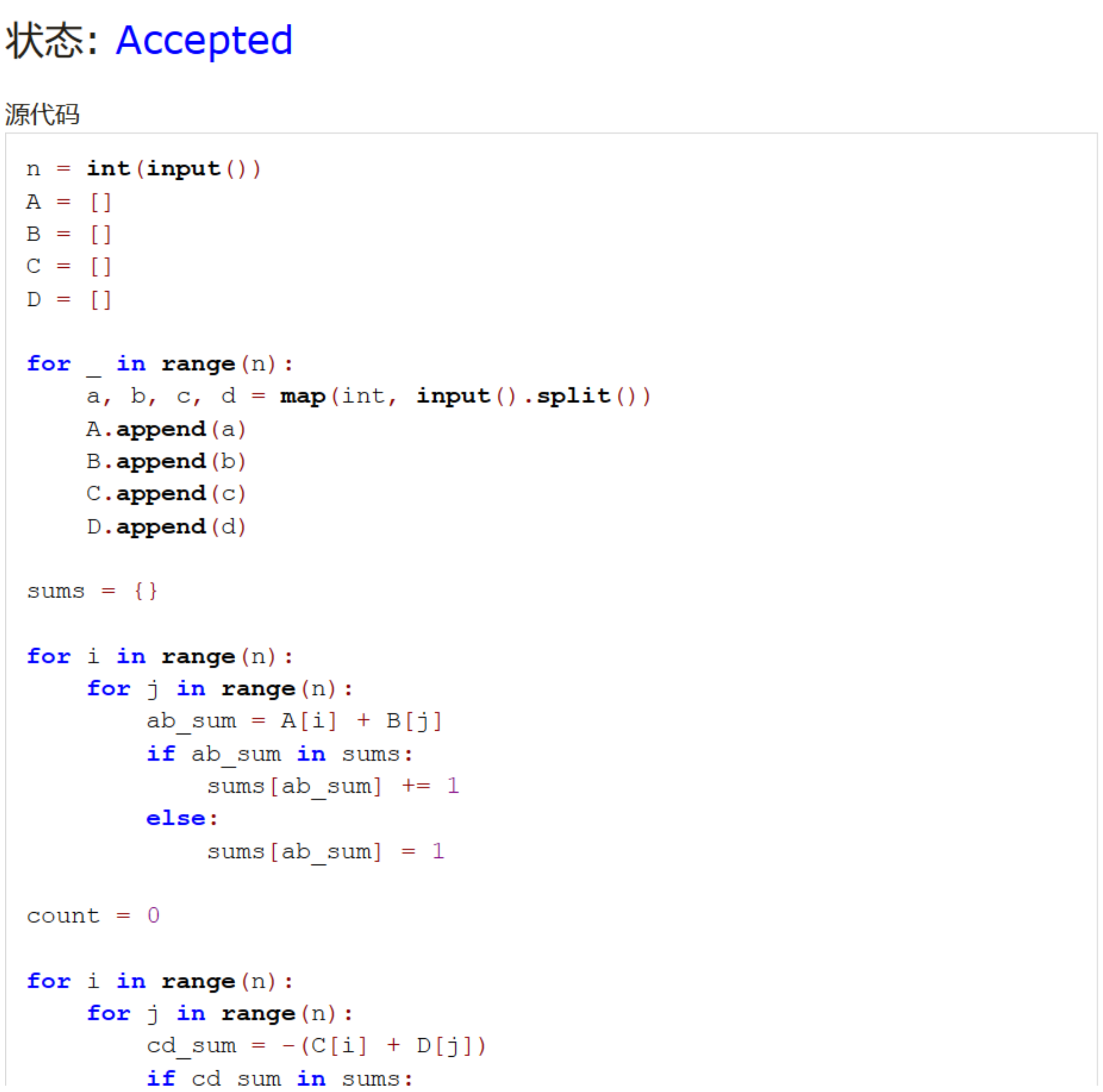
搜索

代码

```
n = int(input())
A = []
B = []
C = []
D = []

for _ in range(n):
    a, b, c, d = map(int, input().split())
    A.append(a)
    B.append(b)
    C.append(c)
    D.append(d)

sums = {}

for i in range(n):
```

```python
    for j in range(n):
        ab_sum = A[i] + B[j]
        if ab_sum in sums:
            sums[ab_sum] += 1
        else:
            sums[ab_sum] = 1

count = 0

for i in range(n):
    for j in range(n):
        cd_sum = -(C[i] + D[j])
        if cd_sum in sums:
            count += sums[cd_sum]

print(count)
```

代码运行截图



状态: Accepted

源代码

```python
n = int(input())
A = []
B = []
C = []
D = []

for _ in range(n):
    a, b, c, d = map(int, input().split())
    A.append(a)
    B.append(b)
    C.append(c)
    D.append(d)

sums = {}

for i in range(n):
    for j in range(n):
        ab_sum = A[i] + B[j]
        if ab_sum in sums:
            sums[ab_sum] += 1
        else:
            sums[ab_sum] = 1

count = 0

for i in range(n):
    for j in range(n):
        cd_sum = -(C[i] + D[j])
        if cd_sum in sums:
```

04089: 电话号码

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

Trie 数据结构可能需要自学下。

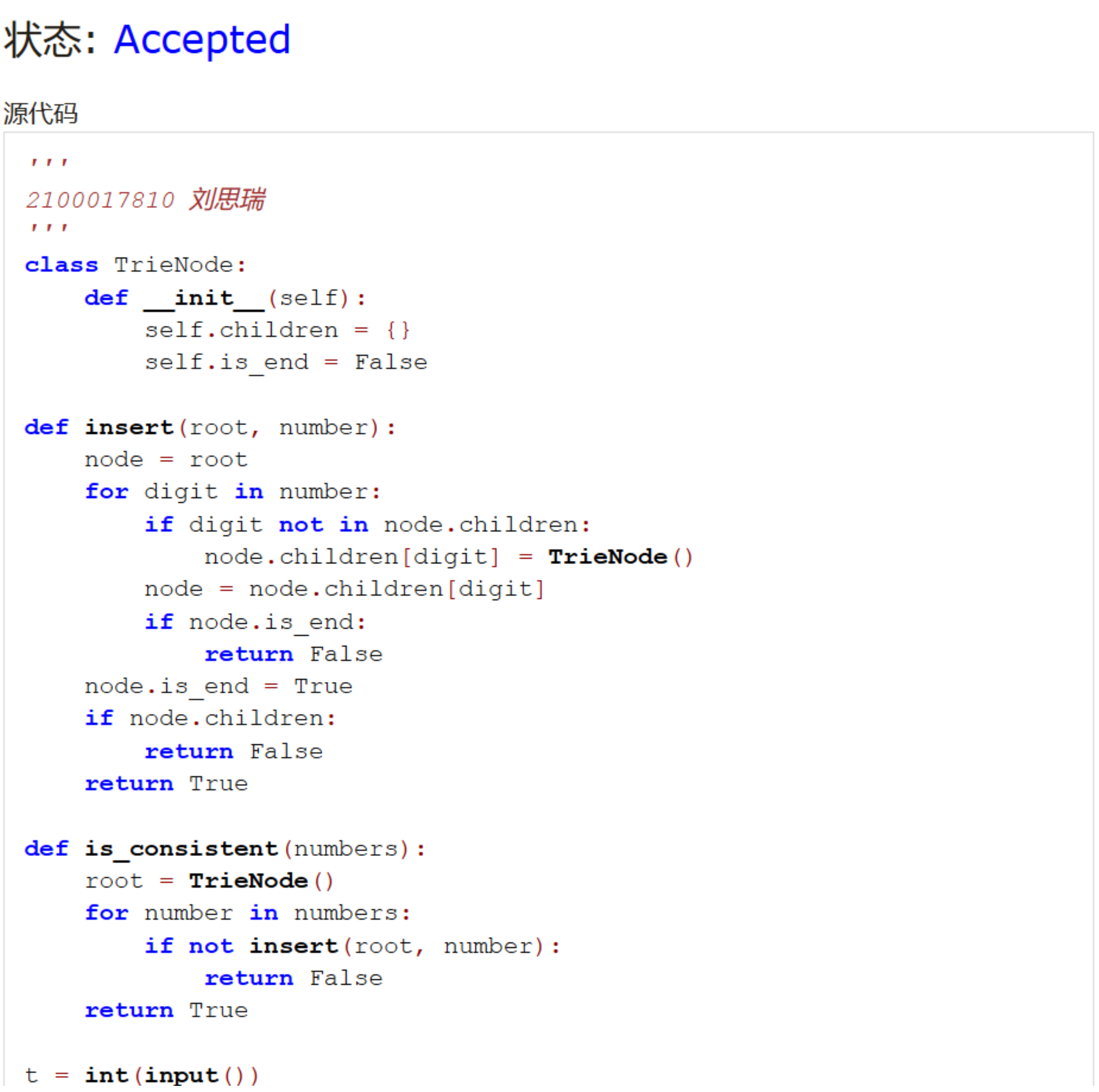
思路：

tire

代码

```python
'''
2100017810 刘思瑞
'''
class TrieNode:
    def __init__(self):
        self.children = {}
        self.is_end = False

def insert(root, number):
    node = root
    for digit in number:
        if digit not in node.children:
            node.children[digit] = TrieNode()
        node = node.children[digit]
        if node.is_end:
            return False
    node.is_end = True
    if node.children:
        return False
    return True

def is_consistent(numbers):
    root = TrieNode()
    for number in numbers:
        if not insert(root, number):
            return False
    return True

t = int(input())

for _ in range(t):
    n = int(input())
    numbers = [input().strip() for _ in range(n)]
    if is_consistent(numbers):
        print("YES")
    else:
        print("NO")
```

代码运行截图



状态: Accepted

源代码

```python
'''
2100017810 刘思瑞
'''
class TrieNode:
    def __init__(self):
        self.children = {}
        self.is_end = False

def insert(root, number):
    node = root
    for digit in number:
        if digit not in node.children:
            node.children[digit] = TrieNode()
        node = node.children[digit]
        if node.is_end:
            return False
    node.is_end = True
    if node.children:
        return False
    return True

def is_consistent(numbers):
    root = TrieNode()
    for number in numbers:
        if not insert(root, number):
            return False
    return True

t = int(input())
```

04082: 树的镜面映射

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

思路：

代码

```
#

```

代码运行截图 （AC代码截图，至少包含有"Accepted"）

2. 学习总结和收获

这周期中周没来得及做太多