Assignment #B: 图论和树算

Updated 1709 GMT+8 Apr 28, 2024

2024 spring, Complied by ==王一粟 经济学院==

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

==（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

28170: 算鹰

dfs, http://cs101.openjudge.cn/practice/28170/

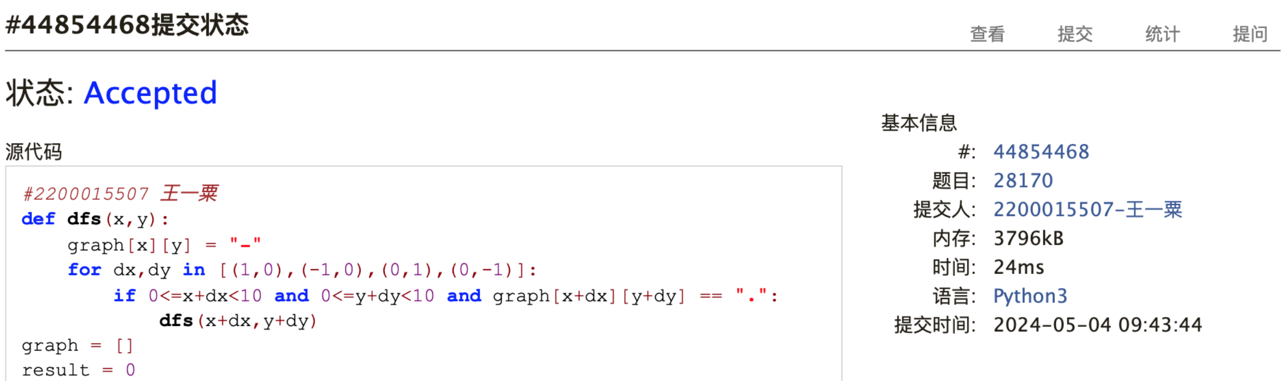
思路：最基本的dfs

耗时：10min

代码

```python
#2200015507 王一粟
def dfs(x,y):
    graph[x][y] = "-"
    for dx,dy in [(1,0),(-1,0),(0,1),(0,-1)]:
        if 0<=x+dx<10 and 0<=y+dy<10 and graph[x+dx][y+dy] == ".":
            dfs(x+dx,y+dy)
graph = []
result = 0
for i in range(10):
    graph.append(list(input()))
for i in range(10):
    for j in range(10):
        if graph[i][j] == ".":
            result += 1
            dfs(i,j)
print(result)
```

代码运行截图 ==（至少包含有"Accepted"）==



#44854468提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```
#2200015507 王一粟
def dfs(x,y):
    graph[x][y] = "-"
    for dx,dy in [(1,0),(-1,0),(0,1),(0,-1)]:
        if 0<=x+dx<10 and 0<=y+dy<10 and graph[x+dx][y+dy] == ".":
            dfs(x+dx,y+dy)
graph = []
result = 0
```

基本信息

#:	44854468
题目:	28170
提交人:	2200015507-王一粟
内存:	3796kB
时间:	24ms
语言:	Python3
提交时间:	2024-05-04 09:43:44

02754: 八皇后

dfs, http://cs101.openjudge.cn/practice/02754/

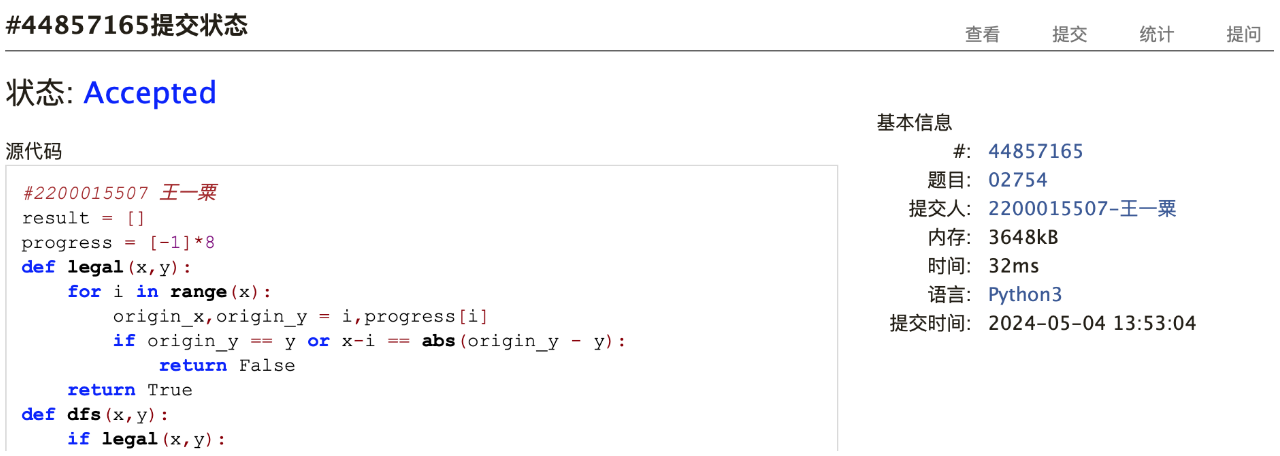
思路：dfs，确认合法往下一行的所有列做调试，调试结束去掉当前的进程。当所有棋子全部落定，记录答案。最后排序即可

耗时：20min

代码

```
#2200015507 王一粟
result = []
progress = [-1]*8
def legal(x,y):
    for i in range(x):
        origin_x,origin_y = i,progress[i]
        if origin_y == y or x-i == abs(origin_y - y):
            return False
    return True
def dfs(x,y):
    if legal(x,y):
        progress[x] = y
        if x == 7:
                result.append(int("".join(str(i+1) for i in progress)))
                progress[x] = -1
            else:
                for next_y in range(8):
                    dfs(x + 1, next_y)
                progress[x] = -1
for y in range(8):
    dfs(0,y)
result.sort()
for _ in range(int(input())):
    print(result[int(input())-1])
```

代码运行截图 ==（至少包含有"Accepted"）==



#44857165提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```
#2200015507 王一粟
result = []
progress = [-1]*8
def legal(x,y):
    for i in range(x):
        origin_x,origin_y = i,progress[i]
        if origin_y == y or x-i == abs(origin_y - y):
            return False
    return True
def dfs(x,y):
    if legal(x,y):
```

基本信息

#:	44857165
题目:	02754
提交人:	2200015507-王一粟
内存:	3648kB
时间:	32ms
语言:	Python3
提交时间:	2024-05-04 13:53:04

03151: Pots

bfs, http://cs101.openjudge.cn/practice/03151/

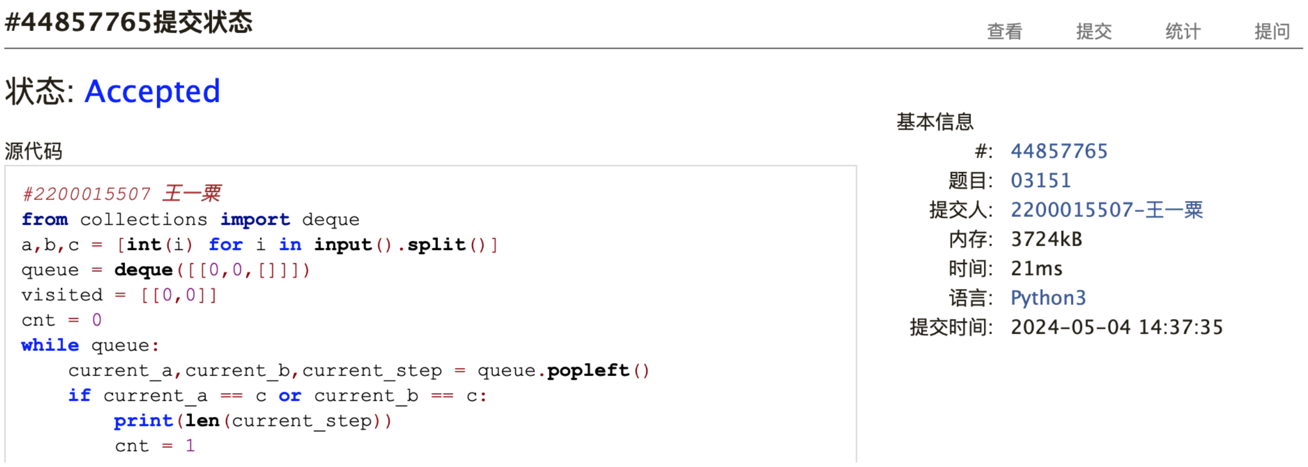
思路：bfs，考虑每一步基于现在pot中的水量可能的操作。用visited存储已经访问过的情况，避免重复访问

耗时：30min

代码

```
#2200015507 王一粟
from collections import deque
a,b,c = [int(i) for i in input().split()]
queue = deque([[0,0,[]]])
visited = [[0,0]]
cnt = 0
while queue:
    current_a,current_b,current_step = queue.popleft()
    if current_a == c or current_b == c:
        print(len(current_step))
        cnt = 1
        for element in current_step:
            print(element)
        break
    if current_a < a and [a,current_b] not in visited:
        queue.append([a,current_b,current_step+["FILL(1)"]])
        visited.append([a,current_b])
    if current_b < b and [current_a,b] not in visited:
        queue.append([current_a,b,current_step+["FILL(2)"]])
        visited.append([current_a,b])
    if current_a < a and current_b > 0:
        next_a = min(a,current_a+current_b)
        next_b = current_a+current_b - next_a
        if [next_a,next_b] not in visited:
            queue.append([next_a,next_b,current_step+["POUR(2,1)"]])
            visited.append([next_a,next_b])
    if current_b < b and current_a > 0:
        next_b = min(b,current_a+current_b)
        next_a = current_a + current_b - next_b
        if [next_a, next_b] not in visited:
            queue.append([next_a,next_b,current_step+["POUR(1,2)"]])
            visited.append([next_a, next_b])
    if current_b > 0 and [current_a,0] not in visited:
        queue.append([current_a,0,current_step+["DROP(2)"]])
        visited.append([current_a,0])
    if current_a > 0:
        queue.append([0,current_b,current_step+["DROP(1)"]])
        visited.append([0,current_b])
if cnt == 0:
    print("impossible")
```

代码运行截图 ==（AC代码截图，至少包含有"Accepted"）==



#44857765提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```
#2200015507 王一粟
from collections import deque
a,b,c = [int(i) for i in input().split()]
queue = deque([[0,0,[]]])
visited = [[0,0]]
cnt = 0
while queue:
    current_a,current_b,current_step = queue.popleft()
    if current_a == c or current_b == c:
        print(len(current_step))
        cnt = 1
```

基本信息

#: 44857765
题目: 03151
提交人: 2200015507-王一粟
内存: 3724kB
时间: 21ms
语言: Python3
提交时间: 2024-05-04 14:37:35

05907: 二叉树的操作

http://cs101.openjudge.cn/practice/05907/

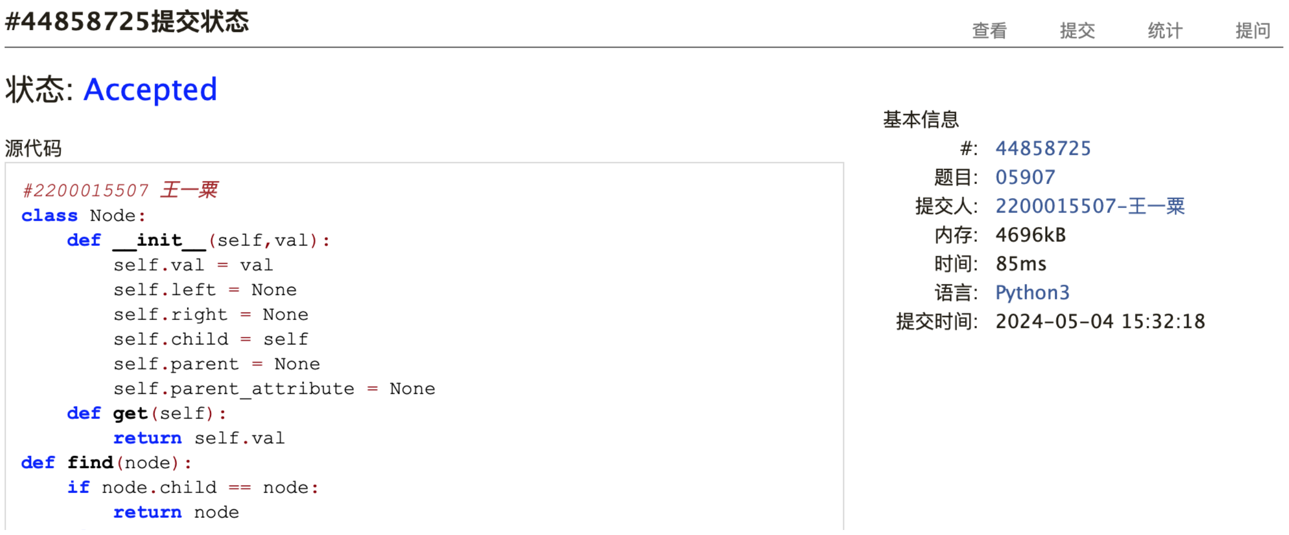
思路：求最左边的结点比较简单，只需要在节点中做维护，每次递归搜索就可以。有一定难度的是交换的时候节点属性的更新，要细致一些

耗时：50min

代码

```
#2200015507 王一粟
class Node:
    def __init__(self,val):
        self.val = val
        self.left = None
        self.right = None
        self.child = self
        self.parent = None
        self.parent_attribute = None
    def get(self):
        return self.val
def find(node):
    if node.child == node:
        return node
    else:
        node.child = find(node.left)
        return node.child
for _ in range(int(input())):
    n,m = [int(i) for i in input().split()]
    node_list = [Node(i) for i in range(n)]
    for i in range(n):
        idx,left_idx,right_idx = [int(i) for i in input().split()]
        if left_idx != -1:
            node_list[idx].left = node_list[left_idx]
            node_list[idx].child = node_list[left_idx].child
            node_list[left_idx].parent = node_list[idx]
            node_list[left_idx].parent_attribute = "left"
        if right_idx != -1:
            node_list[idx].right = node_list[right_idx]
            node_list[right_idx].parent = node_list[idx]
            node_list[right_idx].parent_attribute = "right"
    for i in range(m):
        s = [int(i) for i in input().split()]
        if s[0] == 1:
            use_type,x,y = s
            x = node_list[x]
            y = node_list[y]
            if x.parent_attribute == "left":
                x.parent.left = y
                current_node = x.parent
                x.parent.child = y.child
                while True:
                    if current_node.parent_attribute == "left":
                        current_node.parent.child = y.child
                        current_node = current_node.parent
                    else:
                        break
            else:
                x.parent_right = y
            if y.parent_attribute == "left":
                y.parent.left = x
                current_node = y.parent
                y.parent.child = x.child
                while True:
                    if current_node.parent_attribute == "left":
                        current_node.parent_child = x.child
                        current_node = current_node.parent
                    else:
                        break
            else:
                y.parent_right = x
            x.parent,y.parent = y.parent,x.parent
            x.parent_attribute,y.parent_attribute = y.parent_attribute,x.parent_attribute
        else:
            use_type,x = s
            print(find(node_list[x]).get())
```

代码运行截图 ==（AC代码截图，至少包含有"Accepted"）==



#44858725提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```
#2200015507 王一粟
class Node:
    def __init__(self,val):
        self.val = val
        self.left = None
        self.right = None
        self.child = self
        self.parent = None
        self.parent_attribute = None
    def get(self):
        return self.val
def find(node):
    if node.child == node:
        return node
```

基本信息

#:	44858725
题目:	05907
提交人:	2200015507-王一粟
内存:	4696kB
时间:	85ms
语言:	Python3
提交时间:	2024-05-04 15:32:18

18250: 冰阔落 I

Disjoint set, http://cs101.openjudge.cn/practice/18250/

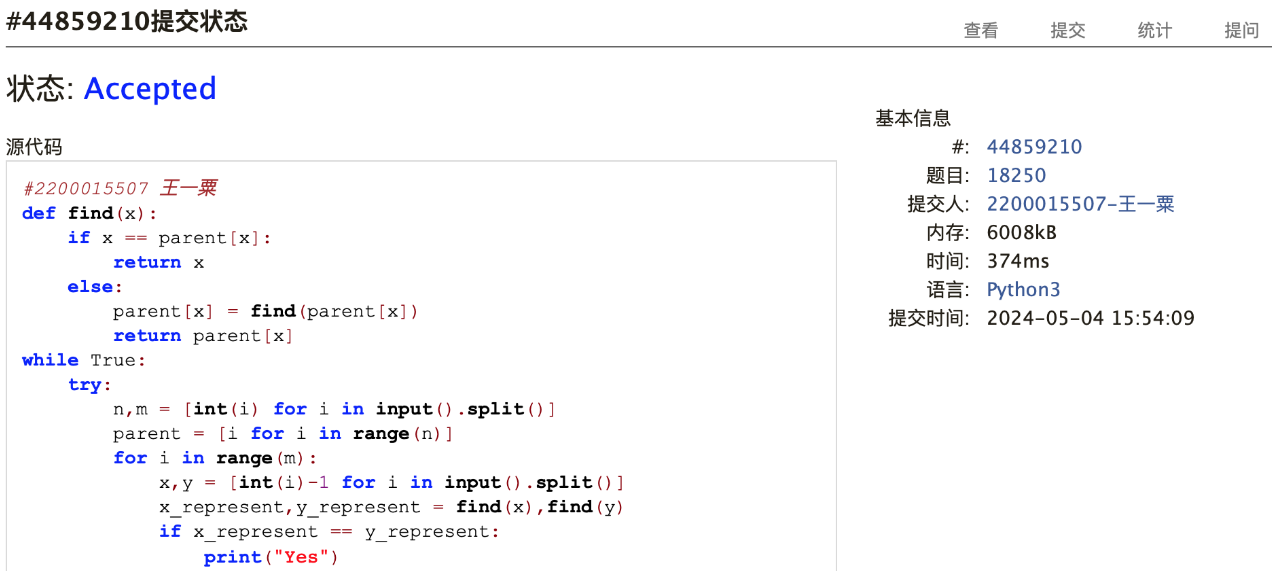
思路：基本的并查集思路

耗时：15min

代码

```
#2200015507 王一粟
def find(x):
    if x == parent[x]:
        return x
    else:
        parent[x] = find(parent[x])
        return parent[x]
while True:
    try:
        n,m = [int(i) for i in input().split()]
        parent = [i for i in range(n)]
        for i in range(m):
            x,y = [int(i)-1 for i in input().split()]
            x_represent,y_represent = find(x),find(y)
            if x_represent == y_represent:
                print("Yes")
            else:
                print("No")
                parent[y_represent] = parent[x_represent]
        result = [i+1 for i in range(n) if i == find(i)]
        print(len(result))
        print(" ".join(str(i) for i in result))
    except:
        break
```

代码运行截图 ==（AC代码截图，至少包含有"Accepted"）==



#44859210提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```
#2200015507 王一粟
def find(x):
    if x == parent[x]:
        return x
    else:
        parent[x] = find(parent[x])
        return parent[x]
while True:
    try:
        n,m = [int(i) for i in input().split()]
        parent = [i for i in range(n)]
        for i in range(m):
            x,y = [int(i)-1 for i in input().split()]
            x_represent,y_represent = find(x),find(y)
            if x_represent == y_represent:
                print("Yes")
```

基本信息

#:	44859210
题目:	18250
提交人:	2200015507-王一粟
内存:	6008kB
时间:	374ms
语言:	Python3
提交时间:	2024-05-04 15:54:09

05443: 兔子与樱花

http://cs101.openjudge.cn/practice/05443/

思路：用node标记了每个位置的前述位置，便于回溯。做的过程中有两点没太注意到，一个是二叉堆的建立过程中，一定要把distance放在前面；第二个是对于distance的更新要在每次插入/更新节点时就完成

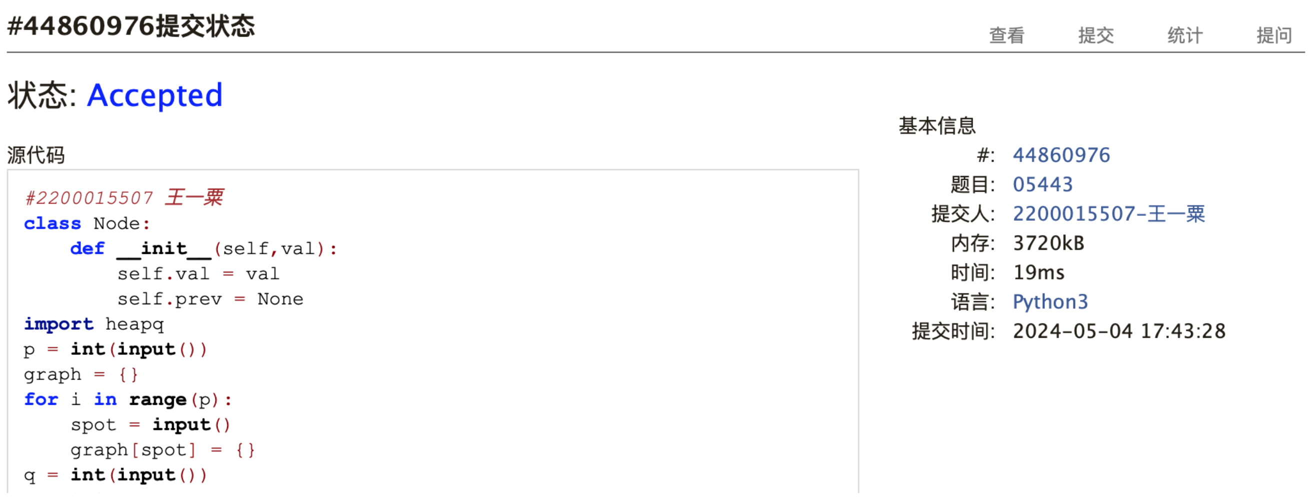
耗时：1.5h

代码

```
#2200015507 王一粟
class Node:
    def __init__(self,val):
        self.val = val
        self.prev = None
```

```
import heapq
p = int(input())
graph = {}
for i in range(p):
    spot = input()
    graph[spot] = {}
q = int(input())
for i in range(q):
    spot1,spot2,distance = input().split()
    dis = int(distance)
    graph[spot1][spot2] = dis
    graph[spot2][spot1] = dis
r = int(input())
for i in range(r):
    start,end = input().split()
    if start == end:
        print(start)
        continue
    distance = {start:0}
    node_dict = {start:Node(start)}
    mylist = []
    heapq.heappush(mylist,[0,start])
    while mylist:
        dist,spot = heapq.heappop(mylist)
        if distance[spot] != dist:
            continue
        distance[spot] = dist
        if spot == end:
            result = end
            current_node = end
            while True:
                if current_node == start:
                    print(result)
                    break
                past_node = node_dict[current_node].prev
                result = past_node + "->(" +str(graph[past_node][current_node]) + ")->" + resu
                current_node = past_node
            break
        for neighbor,inter_dis in graph[spot].items():
            if neighbor in distance and distance[neighbor] < dist+inter_dis:
                continue
            heapq.heappush(mylist,[dist+inter_dis,neighbor])
            distance[neighbor] = dist+inter_dis
            if neighbor not in node_dict:
                new_node = Node(neighbor)
                new_node.prev = spot
                node_dict[neighbor] = new_node
            else:
                node_dict[neighbor].prev = spot
```

代码运行截图 ==（AC代码截图，至少包含有"Accepted"）==


#44860976提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```
#2200015507 王一粟
class Node:
    def __init__(self,val):
        self.val = val
        self.prev = None
import heapq
p = int(input())
graph = {}
for i in range(p):
    spot = input()
    graph[spot] = {}
q = int(input())
```

基本信息

#:	44860976
题目:	05443
提交人:	2200015507-王一粟
内存:	3720kB
时间:	19ms
语言:	Python3
提交时间:	2024-05-04 17:43:28

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。==

趁着五一把之前课件中树的附录和大部分之前未完成的图的讲义学完了

每日选做进度：从4.5补到了4.20