Assignment #B: 图论和树算

Updated 1709 GMT+8 Apr 28, 2024

2024 spring, Complied by Xinjie Song, Phy

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 22H2

Python编程环境：PyCharm 2023.2 (Community Edition)

C/C++编程环境：g++ (x86_64-win32-seh-rev0, Built by MinGW-W64 project) 8.1.0

1. 题目

28170: 算鹰

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

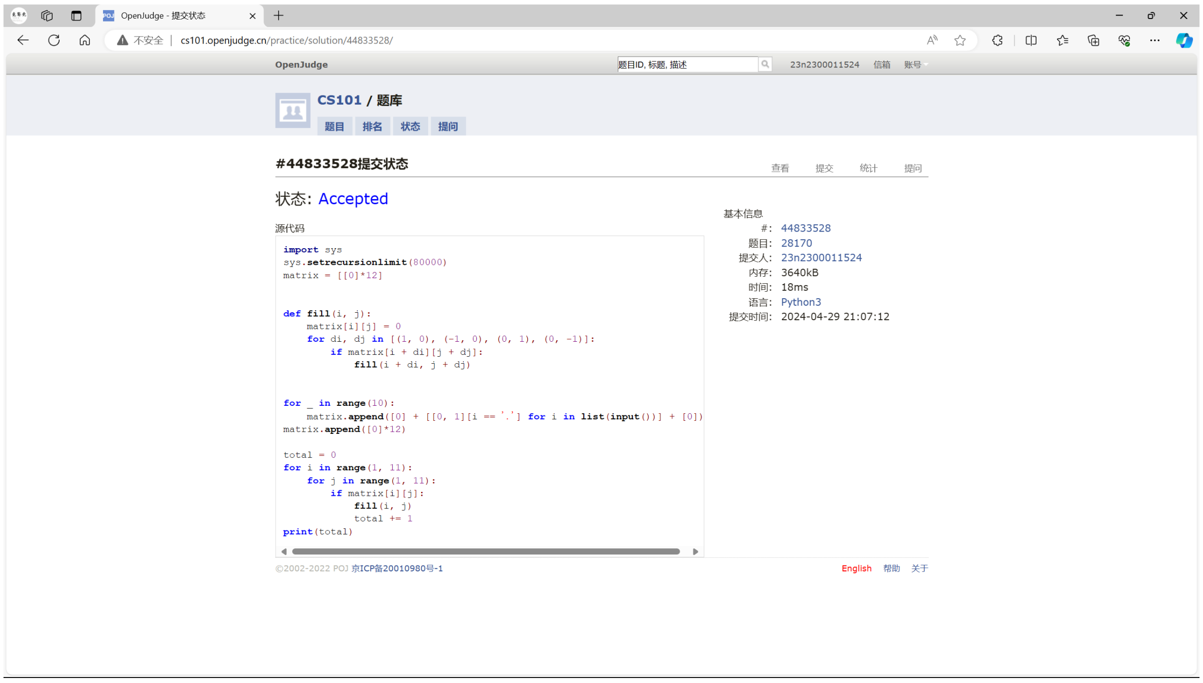
思路：看懂题意就好了......一开始理解错了，案例恰好满足我理解的那种。

代码

```
import sys
sys.setrecursionlimit(80000)
matrix = [[0]*12]


def fill(i, j):
    matrix[i][j] = 0
    for di, dj in [(1, 0), (-1, 0), (0, 1), (0, -1)]:
        if matrix[i + di][j + dj]:
            fill(i + di, j + dj)


for _ in range(10):
    matrix.append([0] + [[0, 1][i == '.'] for i in list(input())] + [0])
```

```python
matrix.append([0]*12)

total = 0
for i in range(1, 11):
    for j in range(1, 11):
        if matrix[i][j]:
            fill(i, j)
            total += 1
print(total)
```

代码运行截图



OpenJudge - 提交状态

cs101.openjudge.cn/practice/solution/44833528/

CS101 / 题库

#44833528提交状态

状态: Accepted

源代码

```python
import sys
sys.setrecursionlimit(80000)
matrix = [[0]*12]


def fill(i, j):
    matrix[i][j] = 0
    for di, dj in [(1, 0), (-1, 0), (0, 1), (0, -1)]:
        if matrix[i + di][j + dj]:
            fill(i + di, j + dj)


for _ in range(10):
    matrix.append([0] + [[0, 1][i == '.'] for i in list(input())] + [0])
matrix.append([0]*12)

total = 0
for i in range(1, 11):
    for j in range(1, 11):
        if matrix[i][j]:
            fill(i, j)
            total += 1
print(total)
```

基本信息

#:	44833528
题目:	28170
提交人:	23n2300011524
内存:	3640kB
时间:	18ms
语言:	Python3
提交时间:	2024-04-29 21:07:12

©2002-2022 POJ 京ICP备20010980号-1

02754: 八皇后

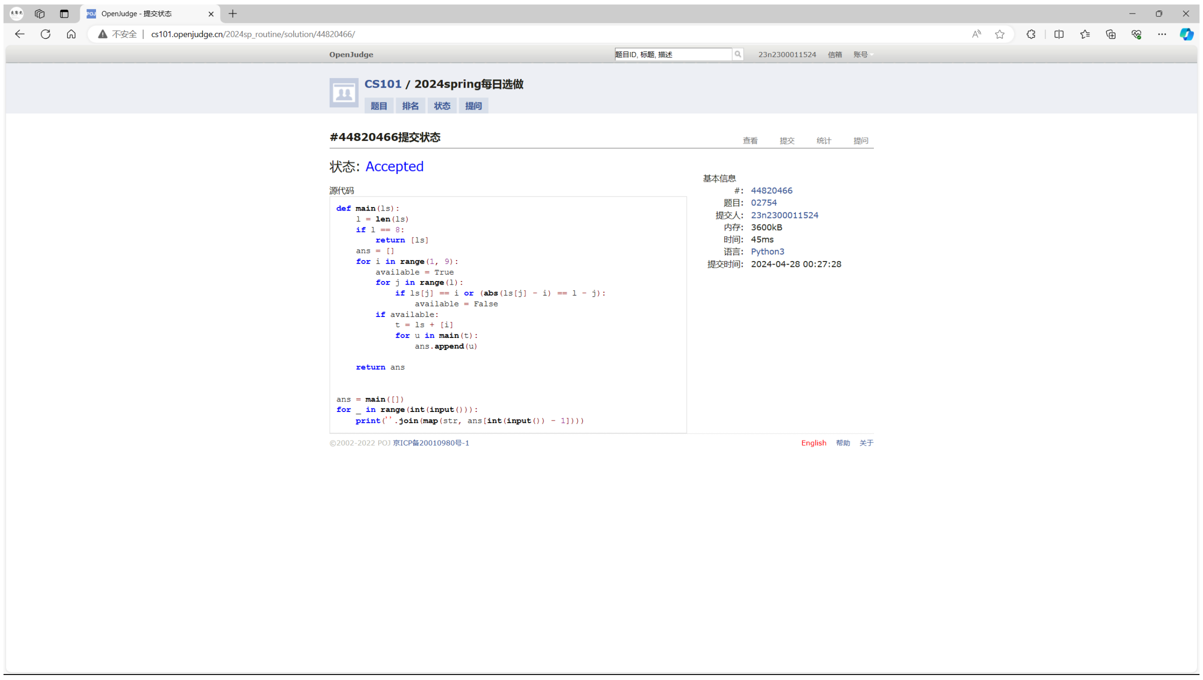
dfs, http://cs101.openjudge.cn/practice/02754/

思路：递归。这次判断合法性换了一种方法，没有建图。

代码

```python
def main(ls):
    l = len(ls)
    if l == 8:
        return [ls]
    ans = []
    for i in range(1, 9):
        available = True
        for j in range(l):
            if ls[j] == i or (abs(ls[j] - i) == l - j):
                available = False
        if available:
                t = ls + [i]
                for u in main(t):
                    ans.append(u)

    return ans


ans = main([])
for _ in range(int(input())):
    print(''.join(map(str, ans[int(input()) - 1])))
```

代码运行截图



OpenJudge

CS101 / 2024spring每日选做

#44820466提交状态

状态: Accepted

源代码

```python
def main(ls):
    l = len(ls)
    if l == 8:
        return [ls]
    ans = []
    for i in range(1, 9):
        available = True
        for j in range(l):
            if ls[j] == i or (abs(ls[j] - i) == l - j):
                available = False
        if available:
            t = ls + [i]
            for u in main(t):
                ans.append(u)

    return ans


ans = main([])
for _ in range(int(input())):
    print(''.join(map(str, ans[int(input()) - 1])))
```

基本信息

#: 44820466
题目: 02754
提交人: 23n2300011524
内存: 3600kB
时间: 45ms
语言: Python3
提交时间: 2024-04-28 00:27:28

©2002-2022 POJ 京ICP备20010980号-1

03151: Pots

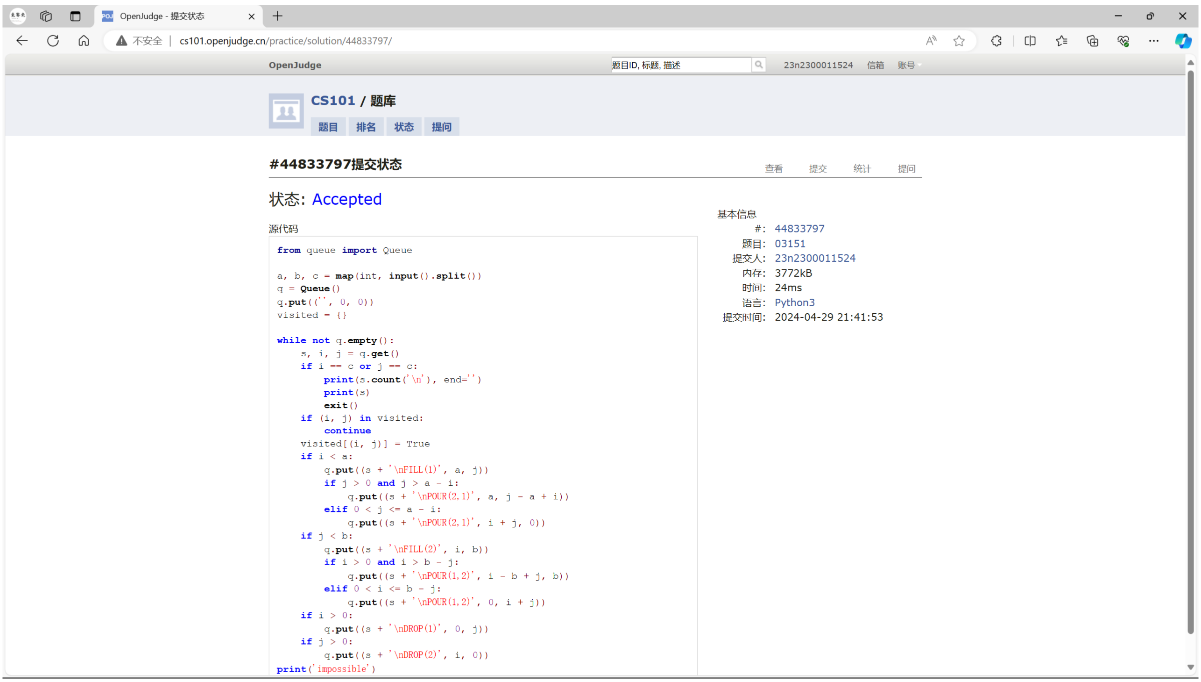
bfs, http://cs101.openjudge.cn/practice/03151/

思路：bfs

代码

```python
from queue import Queue

a, b, c = map(int, input().split())
q = Queue()
q.put(('', 0, 0))
visited = {}

while not q.empty():
    s, i, j = q.get()
    if i == c or j == c:
        print(s.count('\n'), end='')
        print(s)
        exit()
    if (i, j) in visited:
        continue
    visited[(i, j)] = True
    if i < a:
        q.put((s + '\nFILL(1)', a, j))
        if j > 0 and j > a - i:
            q.put((s + '\nPOUR(2,1)', a, j - a + i))
        elif 0 < j <= a - i:
            q.put((s + '\nPOUR(2,1)', i + j, 0))
    if j < b:
        q.put((s + '\nFILL(2)', i, b))
        if i > 0 and i > b - j:
            q.put((s + '\nPOUR(1,2)', i - b + j, b))
        elif 0 < i <= b - j:
            q.put((s + '\nPOUR(1,2)', 0, i + j))
    if i > 0:
        q.put((s + '\nDROP(1)', 0, j))
    if j > 0:
        q.put((s + '\nDROP(2)', i, 0))
print('impossible')
```

代码运行截图



#44833797提交状态

状态: Accepted

源代码

```python
from queue import Queue

a, b, c = map(int, input().split())
q = Queue()
q.put(('', 0, 0))
visited = {}

while not q.empty():
    s, i, j = q.get()
    if i == c or j == c:
        print(s.count('\n'), end='')
        print(s)
        exit()
    if (i, j) in visited:
        continue
    visited[(i, j)] = True
    if i < a:
        q.put((s + '\nFILL(1)', a, j))
        if j > 0 and j > a - i:
            q.put((s + '\nPOUR(2,1)', a, j - a + i))
        elif 0 < j <= a - i:
            q.put((s + '\nPOUR(2,1)', i + j, 0))
    if j < b:
        q.put((s + '\nFILL(2)', i, b))
        if i > 0 and i > b - j:
            q.put((s + '\nPOUR(1,2)', i - b + j, b))
        elif 0 < i <= b - j:
            q.put((s + '\nPOUR(1,2)', 0, i + j))
    if i > 0:
        q.put((s + '\nDROP(1)', 0, j))
    if j > 0:
        q.put((s + '\nDROP(2)', i, 0))
print('impossible')
```

基本信息

#:	44833797
题目:	03151
提交人:	23n2300011524
内存:	3772kB
时间:	24ms
语言:	Python3
提交时间:	2024-04-29 21:41:53

05907: 二叉树的操作

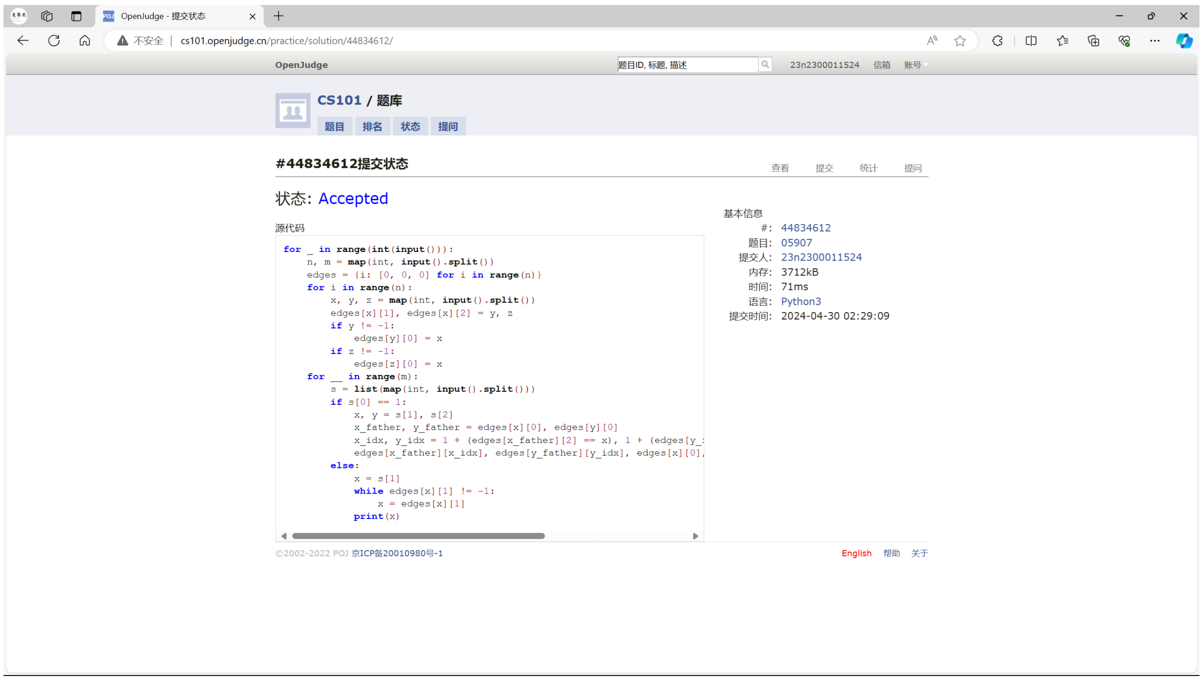
http://cs101.openjudge.cn/practice/05907/

思路：只记录父子关系即可

代码

```python
for _ in range(int(input())):
    n, m = map(int, input().split())
    edges = {i: [0, 0, 0] for i in range(n)}
    for i in range(n):
        x, y, z = map(int, input().split())
        edges[x][1], edges[x][2] = y, z
        if y != -1:
            edges[y][0] = x
        if z != -1:
            edges[z][0] = x
    for __ in range(m):
        s = list(map(int, input().split()))
        if s[0] == 1:
            x, y = s[1], s[2]
            x_father, y_father = edges[x][0], edges[y][0]
            x_idx, y_idx = 1 + (edges[x_father][2] == x), 1 + (edges[y_father][2] == y)
            edges[x_father][x_idx], edges[y_father][y_idx], edges[x][0], edges[y][0] = y, x, y_father, x_father
        else:
            x = s[1]
            while edges[x][1] != -1:
                x = edges[x][1]
            print(x)
```

代码运行截图



#44834612提交状态

状态: Accepted

源代码

```python
for _ in range(int(input())):
    n, m = map(int, input().split())
    edges = {i: [0, 0, 0] for i in range(n)}
    for i in range(n):
        x, y, z = map(int, input().split())
        edges[x][1], edges[x][2] = y, z
        if y != -1:
            edges[y][0] = x
        if z != -1:
            edges[z][0] = x
    for __ in range(m):
        s = list(map(int, input().split()))
        if s[0] == 1:
            x, y = s[1], s[2]
            x_father, y_father = edges[x][0], edges[y][0]
            x_idx, y_idx = 1 + (edges[x_father][2] == x), 1 + (edges[y_
            edges[x_father][x_idx], edges[y_father][y_idx], edges[x][0],
        else:
            x = s[1]
            while edges[x][1] != -1:
                x = edges[x][1]
            print(x)
```

基本信息

#:	44834612
题目:	05907
提交人:	23n2300011524
内存:	3712kB
时间:	71ms
语言:	Python3
提交时间:	2024-04-30 02:29:09

©2002-2022 POJ 京ICP备20010980号-1　English 帮助 关于

18250: 冰阔落 I

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

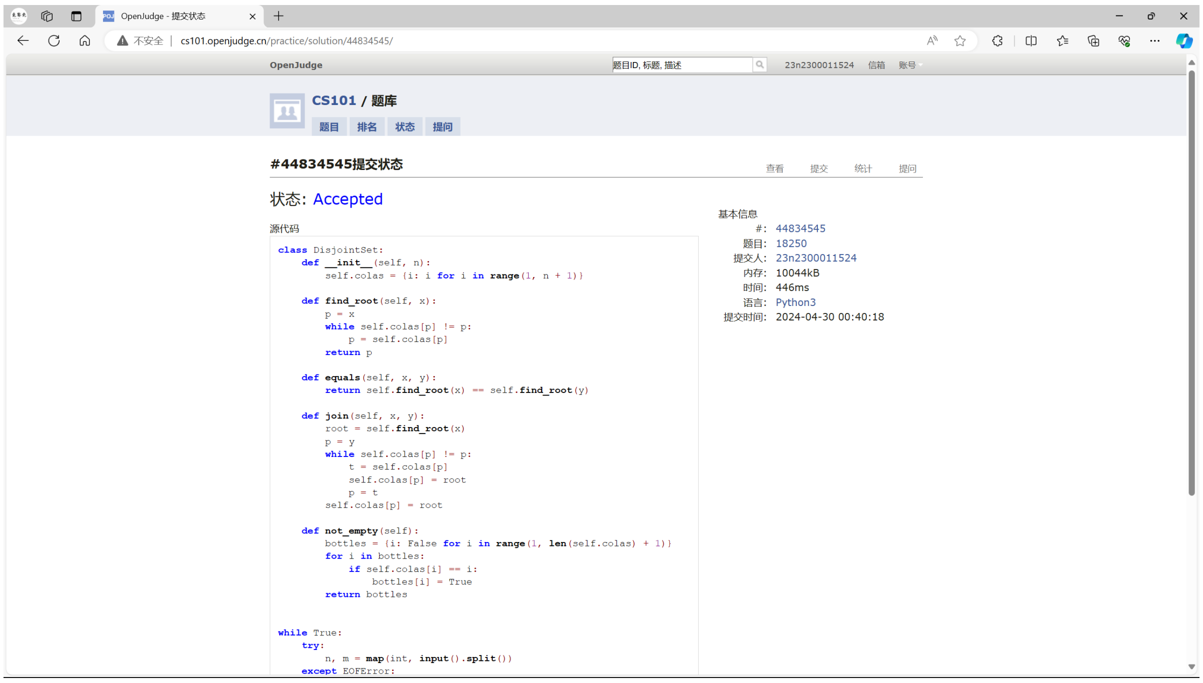
思路：越看越像并查集，一看提示确实是并查集

代码

```
class DisjointSet:
    def __init__(self, n):
        self.colas = {i: i for i in range(1, n + 1)}

    def find_root(self, x):
        p = x
        while self.colas[p] != p:
            p = self.colas[p]
        return p

    def equals(self, x, y):
        return self.find_root(x) == self.find_root(y)

    def join(self, x, y):
        root = self.find_root(x)
        p = y
        while self.colas[p] != p:
            t = self.colas[p]
            self.colas[p] = root
            p = t
        self.colas[p] = root

    def not_empty(self):
        bottles = {i: False for i in range(1, len(self.colas) + 1)}
        for i in bottles:
            if self.colas[i] == i:
                bottles[i] = True
        return bottles


while True:
    try:
        n, m = map(int, input().split())
    except EOFError:
        break
    disjoint_set = DisjointSet(n)
    for _ in range(m):
        x, y = map(int, input().split())
        if disjoint_set.equals(x, y):
            print('Yes')
        else:
            print('No')
            disjoint_set.join(x, y)
    bottles = disjoint_set.not_empty()
```

```python
    ls = []
    for i in bottles:
        if bottles[i]:
            ls.append(str(i))
    print(len(ls))
    print(' '.join(ls))
```

代码运行截图



OpenJudge   23n2300011524   信箱   账号

CS101 / 题库

题目 排名 状态 提问

#44834545提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```python
class DisjointSet:
    def __init__(self, n):
        self.colas = {i: i for i in range(1, n + 1)}

    def find_root(self, x):
        p = x
        while self.colas[p] != p:
            p = self.colas[p]
        return p

    def equals(self, x, y):
        return self.find_root(x) == self.find_root(y)

    def join(self, x, y):
        root = self.find_root(x)
        p = y
        while self.colas[p] != p:
            t = self.colas[p]
            self.colas[p] = root
            p = t
        self.colas[p] = root

    def not_empty(self):
        bottles = {i: False for i in range(1, len(self.colas) + 1)}
        for i in bottles:
            if self.colas[i] == i:
                bottles[i] = True
        return bottles


while True:
    try:
        n, m = map(int, input().split())
    except EOFError:
```

基本信息

#:	44834545
题目:	18250
提交人:	23n2300011524
内存:	10044kB
时间:	446ms
语言:	Python3
提交时间:	2024-04-30 00:40:18

05443: 兔子与樱花

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

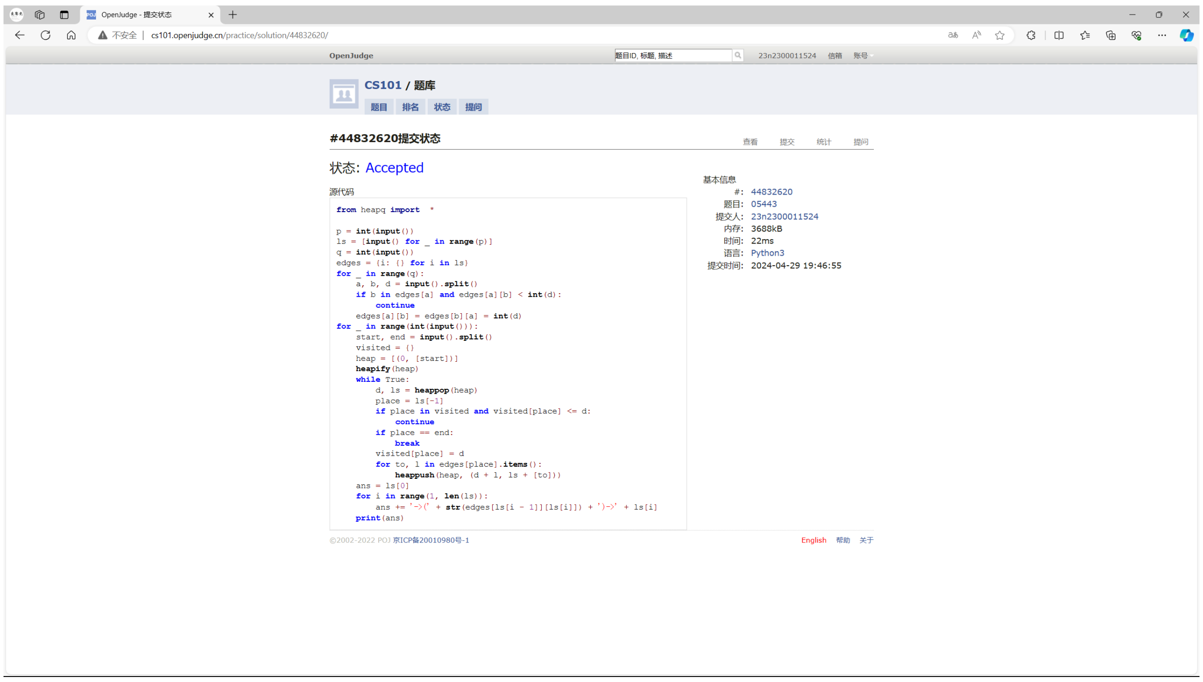
思路：bfs+heap

代码

```python
from heapq import  *

p = int(input())
ls = [input() for _ in range(p)]
q = int(input())
edges = {i: {} for i in ls}
for _ in range(q):
    a, b, d = input().split()
    if b in edges[a] and edges[a][b] < int(d):
        continue
    edges[a][b] = edges[b][a] = int(d)
for _ in range(int(input())):
    start, end = input().split()
    visited = {}
```

```python
        heap = [(0, [start])]
        heapify(heap)
        while True:
            d, ls = heappop(heap)
            place = ls[-1]
            if place in visited and visited[place] <= d:
                continue
            if place == end:
                break
            visited[place] = d
            for to, l in edges[place].items():
                heappush(heap, (d + l, ls + [to]))
        ans = ls[0]
        for i in range(1, len(ls)):
            ans += '->(' + str(edges[ls[i - 1]][ls[i]]) + ')->' + ls[i]
        print(ans)#
```

代码运行截图



#44832620提交状态

状态: Accepted

源代码

```python
from heapq import  *

p = int(input())
ls = [input() for _ in range(p)]
q = int(input())
edges = {i: {} for i in ls}
for _ in range(q):
    a, b, d = input().split()
    if b in edges[a] and edges[a][b] < int(d):
        continue
    edges[a][b] = edges[b][a] = int(d)
for _ in range(int(input())):
    start, end = input().split()
    visited = {}
    heap = [(0, [start])]
    heapify(heap)
    while True:
        d, ls = heappop(heap)
        place = ls[-1]
        if place in visited and visited[place] <= d:
            continue
        if place == end:
            break
        visited[place] = d
        for to, l in edges[place].items():
            heappush(heap, (d + l, ls + [to]))
    ans = ls[0]
    for i in range(1, len(ls)):
        ans += '->(' + str(edges[ls[i - 1]][ls[i]]) + ')->' + ls[i]
    print(ans)
```

基本信息

#:	44832620
题目:	05443
提交人:	23n2300011524
内存:	3688kB
时间:	22ms
语言:	Python3
提交时间:	2024-04-29 19:46:55

2. 学习总结和收获

确实如群里所说，bfs居然能用来解Pots这种题，很有趣。

每日选做在做了。