Assignment #B: 图为主

Updated 2223 GMT+8 Apr 29, 2025

2025 spring, Complied by 颜鼎堃 工学院

> **说明：**
>
> 1. **解题与记录：**
>
>   对于每一个题目，请提供其解题思路（可选），并附上使用Python或C++编写的源代码（确保已在OpenJudge，Codeforces，LeetCode等平台上获得Accepted）。请将这些信息连同显示“Accepted”的截图一起填写到下方的作业模板中。（推荐使用Typora https://typoraio.cn 进行编辑，当然你也可以选择Word。）无论题目是否已通过，请标明每个题目大致花费的时间。
>
> 2. **提交安排：**提交时，请首先上传PDF格式的文件，并将.md或.doc格式的文件作为附件上传至右侧的“作业评论”区。确保你的Canvas账户有一个清晰可见的头像，提交的文件为PDF格式，并且“作业评论”区包含上传的.md或.doc附件。
>
> 3. **延迟提交：**如果你预计无法在截止日期前提交作业，请提前告知具体原因。这有助于我们了解情况并可能为你提供适当的延期或其他帮助。
>
> 请按照上述指导认真准备和提交作业，以保证顺利完成课程要求。

1. 题目

E07218:献给阿尔吉侬的花束

bfs, http://cs101.openjudge.cn/practice/07218/

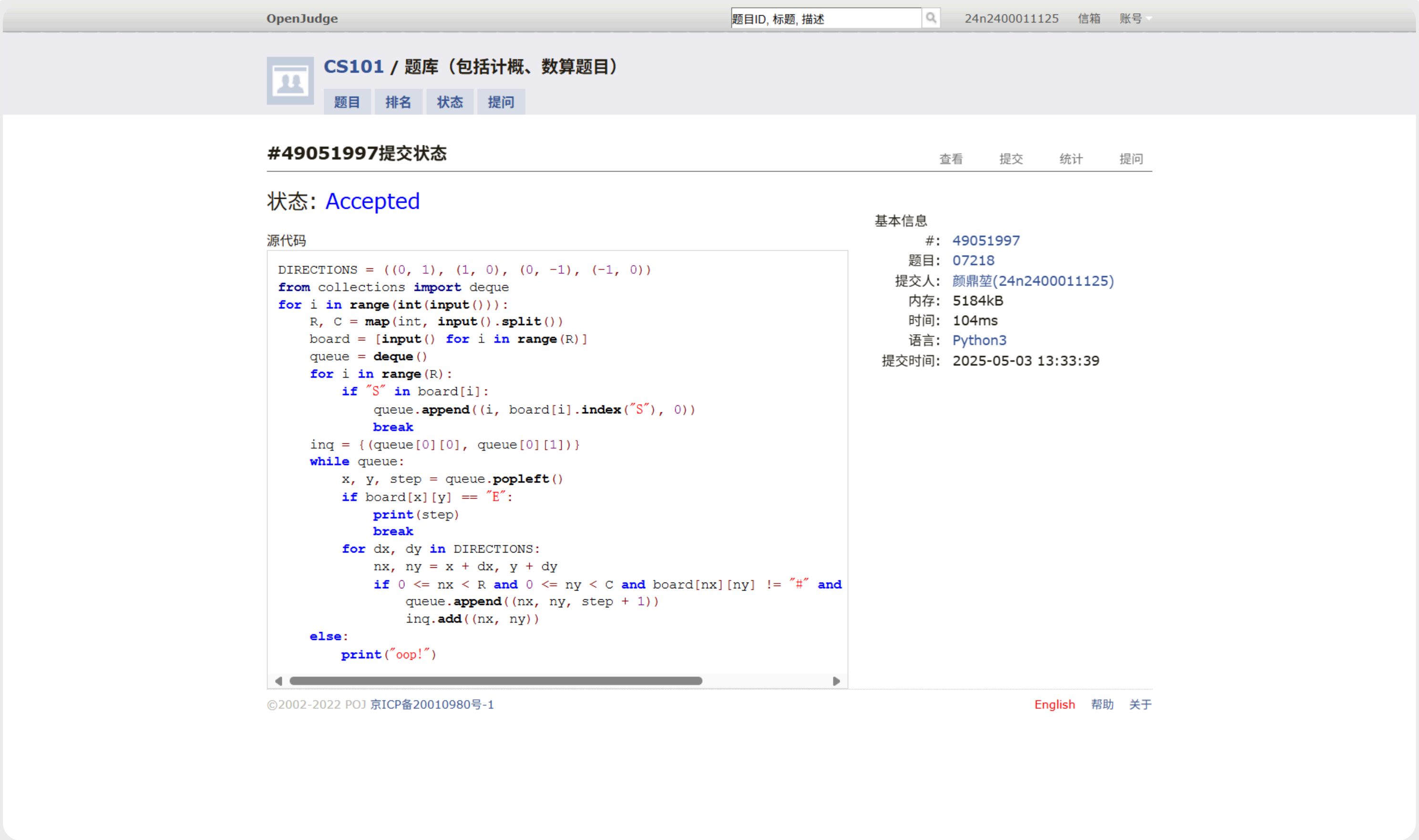
思路：

- 宽搜

代码：

```python
DIRECTIONS = ((0, 1), (1, 0), (0, -1), (-1, 0))
from collections import deque
for i in range(int(input())):
    R, C = map(int, input().split())
    board = [input() for i in range(R)]
    queue = deque()
    for i in range(R):
        if "S" in board[i]:
            queue.append((i, board[i].index("S"), 0))
            break
    inq = {(queue[0][0], queue[0][1])}
    while queue:
        x, y, step = queue.popleft()
        if board[x][y] == "E":
            print(step)
            break
        for dx, dy in DIRECTIONS:
            nx, ny = x + dx, y + dy
            if 0 <= nx < R and 0 <= ny < C and board[nx][ny] != "#" and (nx, ny) not in inq:
                queue.append((nx, ny, step + 1))
                inq.add((nx, ny))
    else:
        print("oop!")

```

代码运行截图 （至少包含有"Accepted"）



#49051997提交状态

状态: Accepted

源代码

```
DIRECTIONS = ((0, 1), (1, 0), (0, -1), (-1, 0))
from collections import deque
for i in range(int(input())):
    R, C = map(int, input().split())
    board = [input() for i in range(R)]
    queue = deque()
    for i in range(R):
        if "S" in board[i]:
            queue.append((i, board[i].index("S"), 0))
            break
    inq = {(queue[0][0], queue[0][1])}
    while queue:
        x, y, step = queue.popleft()
        if board[x][y] == "E":
            print(step)
            break
        for dx, dy in DIRECTIONS:
            nx, ny = x + dx, y + dy
            if 0 <= nx < R and 0 <= ny < C and board[nx][ny] != "#" and
                queue.append((nx, ny, step + 1))
                inq.add((nx, ny))
    else:
        print("oop!")
```

基本信息

#: 49051997
题目: 07218
提交人: 颜鼎堃(24n2400011125)
内存: 5184kB
时间: 104ms
语言: Python3
提交时间: 2025-05-03 13:33:39

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

OJ-07218

M3532.针对图的路径存在性查询I

disjoint set, https://leetcode.cn/problems/path-existence-queries-in-a-graph-i/

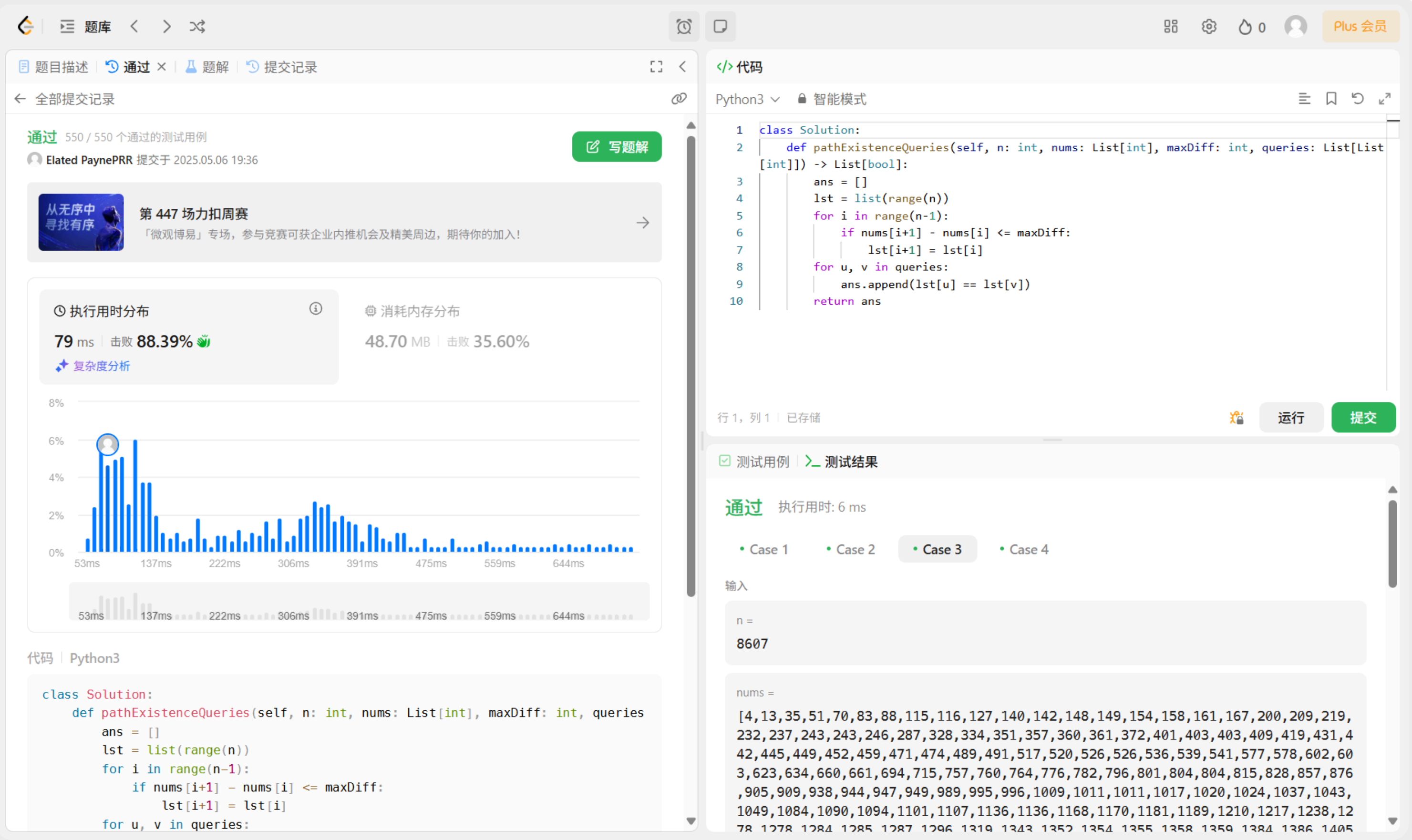
思路：

- 被这题卡了好久
- 上来就想到了并查集，但是没想到数组有序能把读入处理速度降到$O(n)$
- 导致读入写的双循环，后面并查集优化到极致，拼尽全力通过了测试数据433后通不过436了
- 不过也有好处，现在学会并查集路径压缩怎么写了
- 以及，发现海象运算符不能给数组中特定下标的元素赋值

代码：

```python
from typing import *
class Solution:
    def pathExistenceQueries(self, n: int, nums: List[int], maxDiff: int, queries: List[List[int]]) -> List[bool]:
        ans = []
        lst = list(range(n))
        for i in range(n-1):
            if nums[i+1] - nums[i] <= maxDiff:
                lst[i+1] = lst[i]
        for u, v in queries:
            ans.append(lst[u] == lst[v])
        return ans


if __name__ == "__main__":
    sol = Solution()
    print(sol.pathExistenceQueries(n = 2, nums = [1,3], maxDiff = 1, queries = [[0,0],[0,1]]))

```

代码运行截图 （至少包含有"Accepted"）



通过 550 / 550 个通过的测试用例

Elated PaynePRR 提交于 2025.05.06 19:36

79 ms | 击败 88.39%

48.70 MB | 击败 35.60%

代码 | Python3

```python
class Solution:
    def pathExistenceQueries(self, n: int, nums: List[int], maxDiff: int, queries
        ans = []
        lst = list(range(n))
        for i in range(n-1):
            if nums[i+1] - nums[i] <= maxDiff:
                lst[i+1] = lst[i]
        for u, v in queries:
```

```python
class Solution:
    def pathExistenceQueries(self, n: int, nums: List[int], maxDiff: int, queries: List[List[int]]) -> List[bool]:
        ans = []
        lst = list(range(n))
        for i in range(n-1):
            if nums[i+1] - nums[i] <= maxDiff:
                lst[i+1] = lst[i]
        for u, v in queries:
            ans.append(lst[u] == lst[v])
        return ans
```

行 1，列 1 | 已存储

通过 执行用时: 6 ms

输入

n =

8607

nums =

```
[4,13,35,51,70,83,88,115,116,127,140,142,148,149,154,158,161,167,200,209,219,
232,237,243,243,246,287,328,334,351,357,360,361,372,401,403,403,409,419,431,4
42,445,449,452,459,471,474,489,491,517,520,526,526,536,539,541,577,578,602,60
3,623,634,660,661,694,715,757,760,764,776,782,796,801,804,804,815,828,857,876
,905,909,938,944,947,949,989,995,996,1009,1011,1011,1017,1020,1024,1037,1043,
1049,1084,1090,1094,1101,1107,1136,1136,1168,1170,1181,1189,1210,1217,1238,12
78,1278,1284,1285,1287,1296,1319,1343,1352,1354,1355,1358,1359,1384,1386,1405
```

LC-3532

M22528:厚道的调分方法

binary search, http://cs101.openjudge.cn/practice/22528/

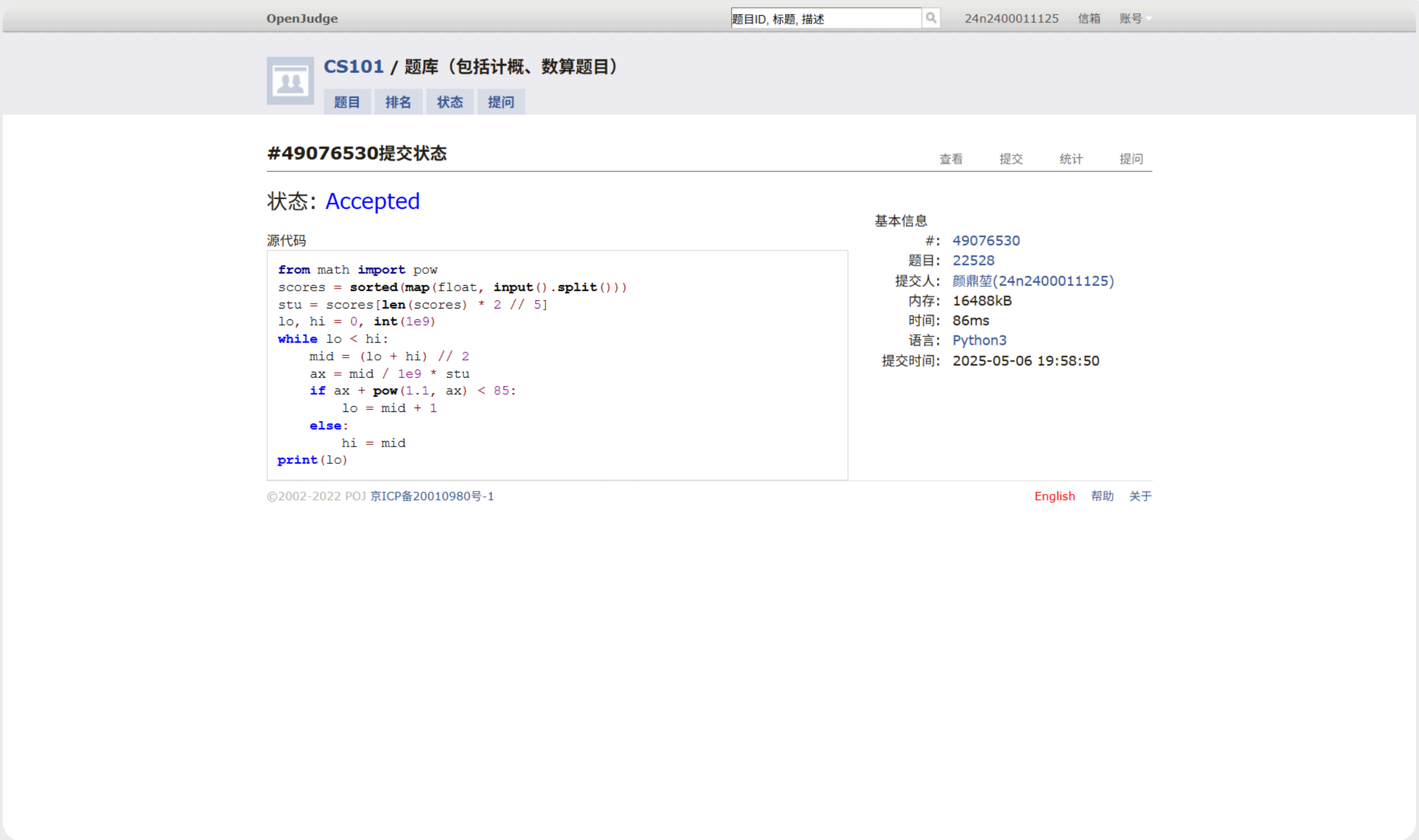
思路：

- 看到浮点数还挺担心的，还好一次过了
- 想起了上学期被一元二次方程求解支配的日子

代码：

```python
from math import pow
scores = sorted(map(float, input().split()))
stu = scores[len(scores) * 2 // 5]
lo, hi = 0, int(1e9)
while lo < hi:
    mid = (lo + hi) // 2
    ax = mid / 1e9 * stu
    if ax + pow(1.1, ax) < 85:
        lo = mid + 1
    else:
        hi = mid
print(lo)
```

代码运行截图 （至少包含有"Accepted"）



#49076530提交状态

状态: Accepted

源代码

```python
from math import pow
scores = sorted(map(float, input().split()))
stu = scores[len(scores) * 2 // 5]
lo, hi = 0, int(1e9)
while lo < hi:
    mid = (lo + hi) // 2
    ax = mid / 1e9 * stu
    if ax + pow(1.1, ax) < 85:
        lo = mid + 1
    else:
        hi = mid
print(lo)
```

基本信息

#:	49076530
题目:	22528
提交人:	颜鼎堃(24n2400011125)
内存:	16488kB
时间:	86ms
语言:	Python3
提交时间:	2025-05-06 19:58:50

©2002-2022 POJ 京ICP备20010980号-1

OJ-22528

Msy382: 有向图判环

dfs, https://sunnywhy.com/sfbj/10/3/382

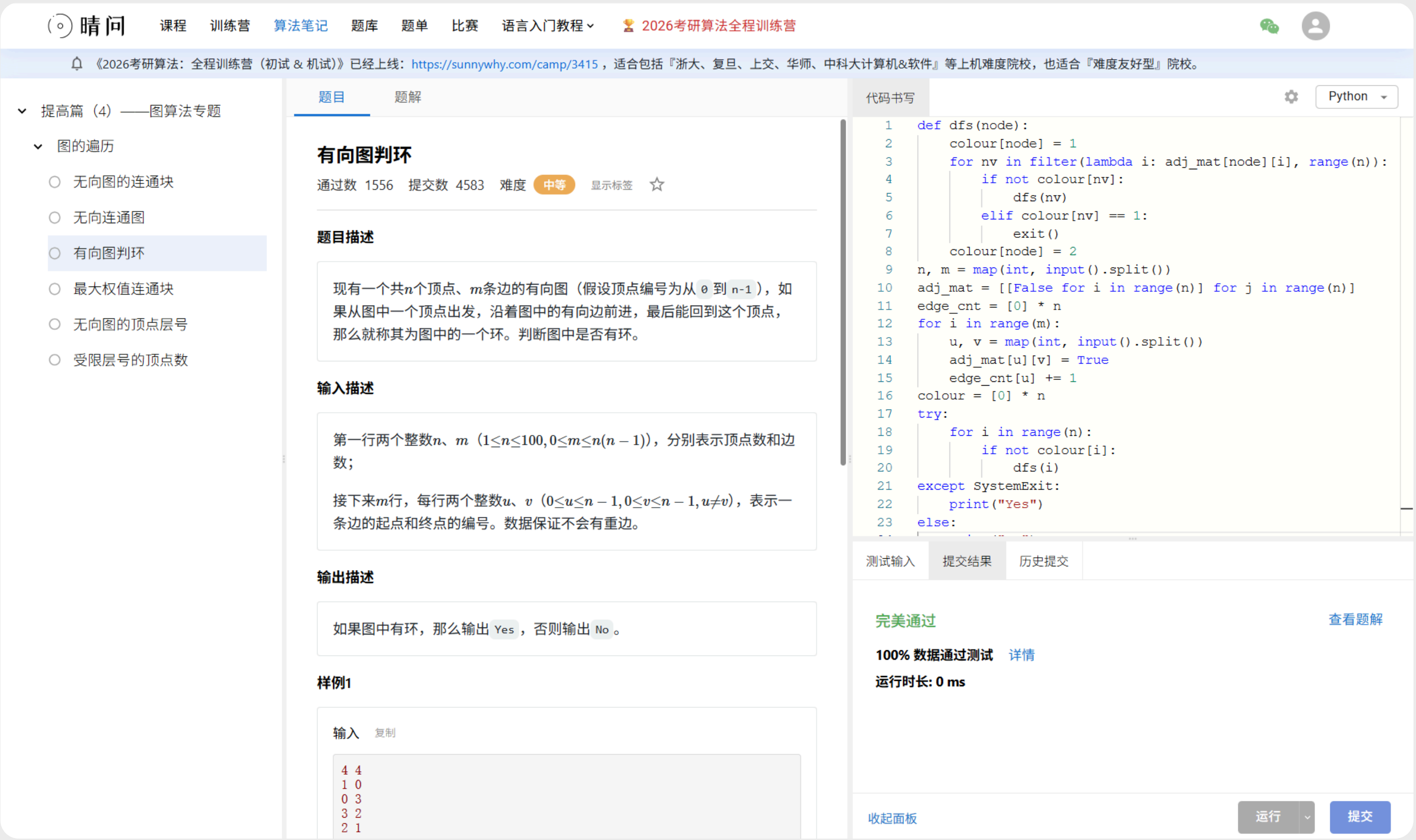
思路：

- 用深搜标记三种状态
- 捕获 `SystemExit` 退出递归

代码：

```python
def dfs(node):
    colour[node] = 1
    for nv in filter(lambda i: adj_mat[node][i], range(n)):
        if not colour[nv]:
            dfs(nv)
        elif colour[nv] == 1:
            exit()
    colour[node] = 2
n, m = map(int, input().split())
adj_mat = [[False for i in range(n)] for j in range(n)]
edge_cnt = [0] * n
for i in range(m):
    u, v = map(int, input().split())
    adj_mat[u][v] = True
    edge_cnt[u] += 1
colour = [0] * n
try:
    for i in range(n):
        if not colour[i]:
            dfs(i)
except SystemExit:
    print("Yes")
else:
    print("No")
```

代码运行截图 （至少包含有"Accepted"）



有向图判环

通过数 1556 提交数 4583 难度 中等 显示标签 ☆

题目描述

现有一个共n个顶点、m条边的有向图（假设顶点编号为从`0`到`n-1`），如果从图中一个顶点出发，沿着图中的有向边前进，最后能回到这个顶点，那么就称其为图中的一个环。判断图中是否有环。

输入描述

第一行两个整数n、m（$1 \le n \le 100, 0 \le m \le n(n-1)$），分别表示顶点数和边数；

接下来m行，每行两个整数u、v（$0 \le u \le n-1, 0 \le v \le n-1, u \ne v$），表示一条边的起点和终点的编号。数据保证不会有重边。

输出描述

如果图中有环，那么输出`Yes`，否则输出`No`。

样例1

输入 复制

```
4 4
1 0
0 3
3 2
2 1
```

```python
def dfs(node):
    colour[node] = 1
    for nv in filter(lambda i: adj_mat[node][i], range(n)):
        if not colour[nv]:
            dfs(nv)
        elif colour[nv] == 1:
            exit()
    colour[node] = 2
n, m = map(int, input().split())
adj_mat = [[False for i in range(n)] for j in range(n)]
edge_cnt = [0] * n
for i in range(m):
    u, v = map(int, input().split())
    adj_mat[u][v] = True
    edge_cnt[u] += 1
colour = [0] * n
try:
    for i in range(n):
        if not colour[i]:
            dfs(i)
except SystemExit:
    print("Yes")
else:
```

完美通过 查看题解

100% 数据通过测试 详情

运行时长: 0 ms

SY-382

M05443:兔子与樱花

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

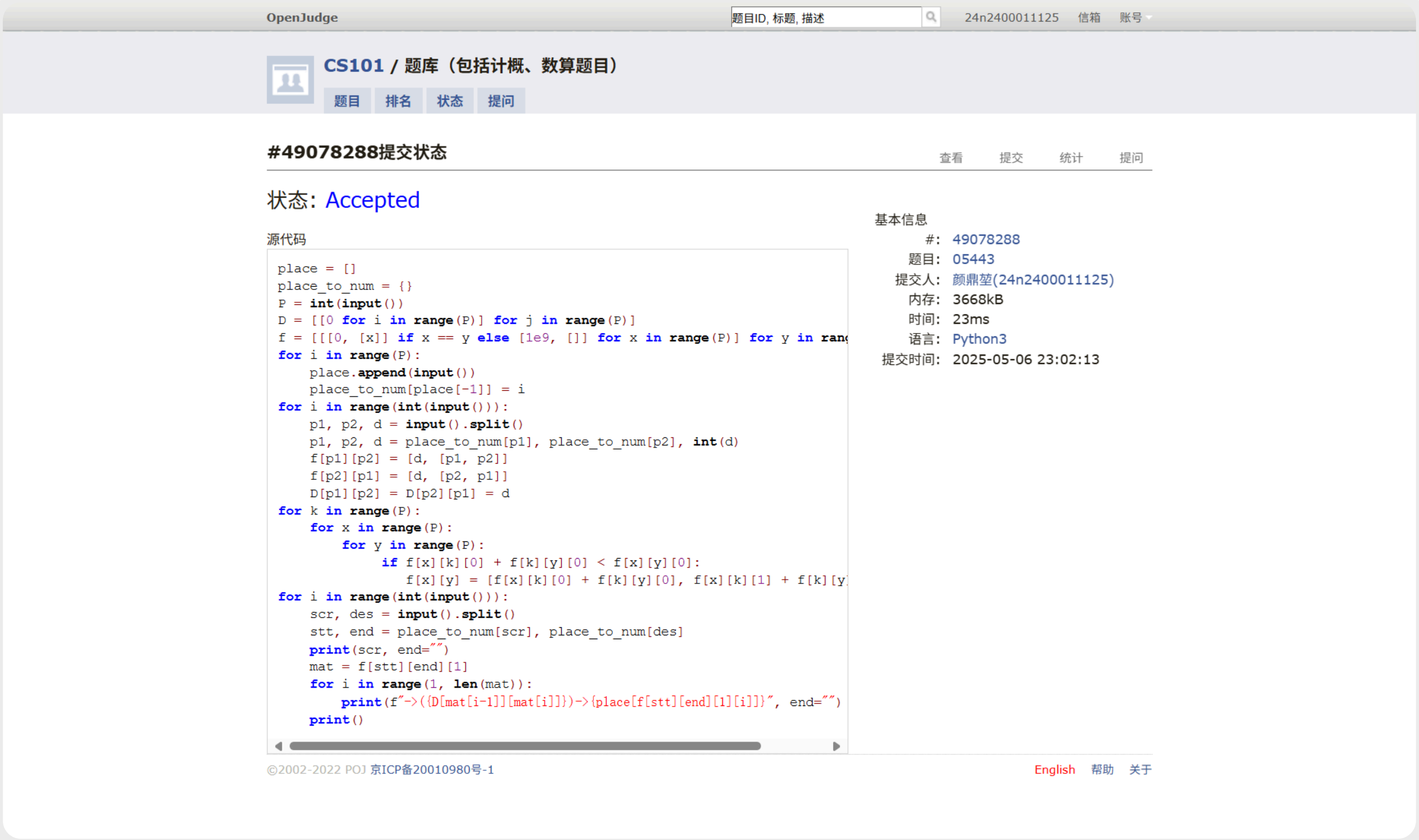
思路：

- 太好了是全源最短路我们没救了
- 去进行了一些学习，弗洛伊德算法写起来确实快，小规模的图也够用了
- 不过也顺便复习了迪杰斯特拉算法

代码：

```python
place = []
place_to_num = {}
P = int(input())
D = [[0 for i in range(P)] for j in range(P)]
f = [[[0, [x]] if x == y else [1e9, []] for x in range(P)] for y in range(P)]
for i in range(P):
    place.append(input())
    place_to_num[place[-1]] = i
for i in range(int(input())):
    p1, p2, d = input().split()
    p1, p2, d = place_to_num[p1], place_to_num[p2], int(d)
    f[p1][p2] = [d, [p1, p2]]
    f[p2][p1] = [d, [p2, p1]]
    D[p1][p2] = D[p2][p1] = d
for k in range(P):
    for x in range(P):
        for y in range(P):
            if f[x][k][0] + f[k][y][0] < f[x][y][0]:
                f[x][y] = [f[x][k][0] + f[k][y][0], f[x][k][1] + f[k][y][1][1:]]
for i in range(int(input())):
    scr, des = input().split()
    stt, end = place_to_num[scr], place_to_num[des]
    print(scr, end="")
    mat = f[stt][end][1]
    for i in range(1, len(mat)):
        print(f"->({D[mat[i-1]][mat[i]]})->{place[f[stt][end][1][i]]}", end="")
    print()

```

代码运行截图 （至少包含有"Accepted"）



#49078288提交状态

状态: Accepted

源代码

```
place = []
place_to_num = {}
P = int(input())
D = [[0 for i in range(P)] for j in range(P)]
f = [[[0, [x]] if x == y else [1e9, []] for x in range(P)] for y in rang
for i in range(P):
    place.append(input())
    place_to_num[place[-1]] = i
for i in range(int(input())):
    p1, p2, d = input().split()
    p1, p2, d = place_to_num[p1], place_to_num[p2], int(d)
    f[p1][p2] = [d, [p1, p2]]
    f[p2][p1] = [d, [p2, p1]]
    D[p1][p2] = D[p2][p1] = d
for k in range(P):
    for x in range(P):
        for y in range(P):
            if f[x][k][0] + f[k][y][0] < f[x][y][0]:
                f[x][y] = [f[x][k][0] + f[k][y][0], f[x][k][1] + f[k][y
for i in range(int(input())):
    scr, des = input().split()
    stt, end = place_to_num[scr], place_to_num[des]
    print(scr, end="")
    mat = f[stt][end][1]
    for i in range(1, len(mat)):
        print(f"->({D[mat[i-1]][mat[i]]})->{place[f[stt][end][1][i]]}", end="")
    print()
```

基本信息

#: 49078288
题目: 05443
提交人: 颜鼎堃(24n2400011125)
内存: 3668kB
时间: 23ms
语言: Python3
提交时间: 2025-05-06 23:02:13

©2002-2022 POJ 京ICP备20010980号-1

English 帮助 关于

OJ-05443

T28050: 骑士周游

dfs, http://cs101.openjudge.cn/practice/28050/

思路：

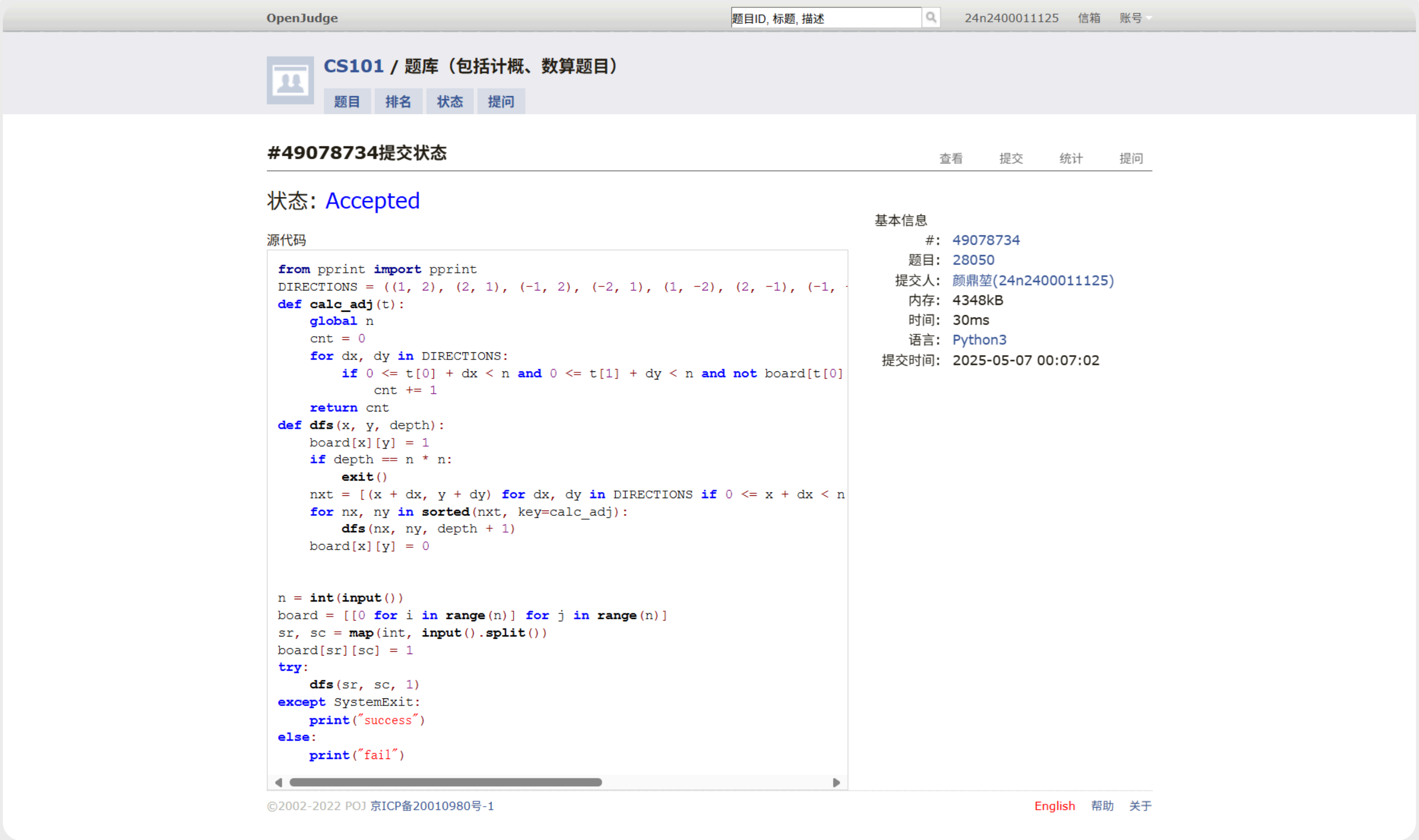
- 原来这就是传说中的启发式搜索
- 试了一下，找最少的和找最多的时间差了不是一星半点

代码：

```Python
DIRECTIONS = ((1, 2), (2, 1), (-1, 2), (-2, 1), (1, -2), (2, -1), (-1, -2), (-2, -1))
def calc_adj(t):
    global n
    cnt = 0
    for dx, dy in DIRECTIONS:
        if 0 <= t[0] + dx < n and 0 <= t[1] + dy < n and not board[t[0] + dx][t[1] + dy]:
            cnt += 1
    return cnt
def dfs(x, y, depth):
    board[x][y] = 1
    if depth == n * n:
        exit()
    nxt = [(x + dx, y + dy) for dx, dy in DIRECTIONS if 0 <= x + dx < n and 0 <= y + dy < n and not board[x + dx][y + dy]]
    for nx, ny in sorted(nxt, key=calc_adj):
        dfs(nx, ny, depth + 1)
    board[x][y] = 0

```

```python

n = int(input())
board = [[0 for i in range(n)] for j in range(n)]
sr, sc = map(int, input().split())
board[sr][sc] = 1
try:
    dfs(sr, sc, 1)
except SystemExit:
    print("success")
else:
    print("fail")

```

代码运行截图（至少包含有"Accepted"）



#49078734提交状态

状态: Accepted

源代码

```python
from pprint import pprint
DIRECTIONS = ((1, 2), (2, 1), (-1, 2), (-2, 1), (1, -2), (2, -1), (-1, 
def calc_adj(t):
    global n
    cnt = 0
    for dx, dy in DIRECTIONS:
        if 0 <= t[0] + dx < n and 0 <= t[1] + dy < n and not board[t[0]
            cnt += 1
    return cnt
def dfs(x, y, depth):
    board[x][y] = 1
    if depth == n * n:
        exit()
    nxt = [(x + dx, y + dy) for dx, dy in DIRECTIONS if 0 <= x + dx < n
    for nx, ny in sorted(nxt, key=calc_adj):
        dfs(nx, ny, depth + 1)
    board[x][y] = 0


n = int(input())
board = [[0 for i in range(n)] for j in range(n)]
sr, sc = map(int, input().split())
board[sr][sc] = 1
try:
    dfs(sr, sc, 1)
except SystemExit:
    print("success")
else:
    print("fail")
```

基本信息

#: 49078734
题目: 28050
提交人: 颜鼎堃(24n2400011125)
内存: 4348kB
时间: 30ms
语言: Python3
提交时间: 2025-05-07 00:07:02

OJ-28050

2. 学习总结和收获

如果发现作业题目相对简单，有否寻找额外的练习题目，如“数算2025spring每日选做”、LeetCode、Codeforces、洛谷等网站上的题目。

很多算法多少有点忘了，还得复习

以及并查集似乎在做一些图的题目时可以用来逃课

比如OpenJudge - 02815:城堡问题

直接建一个$m \cdot n$大小的并查集就好，不需要搜索了