1. 题目

E07218:献给阿尔吉侬的花束

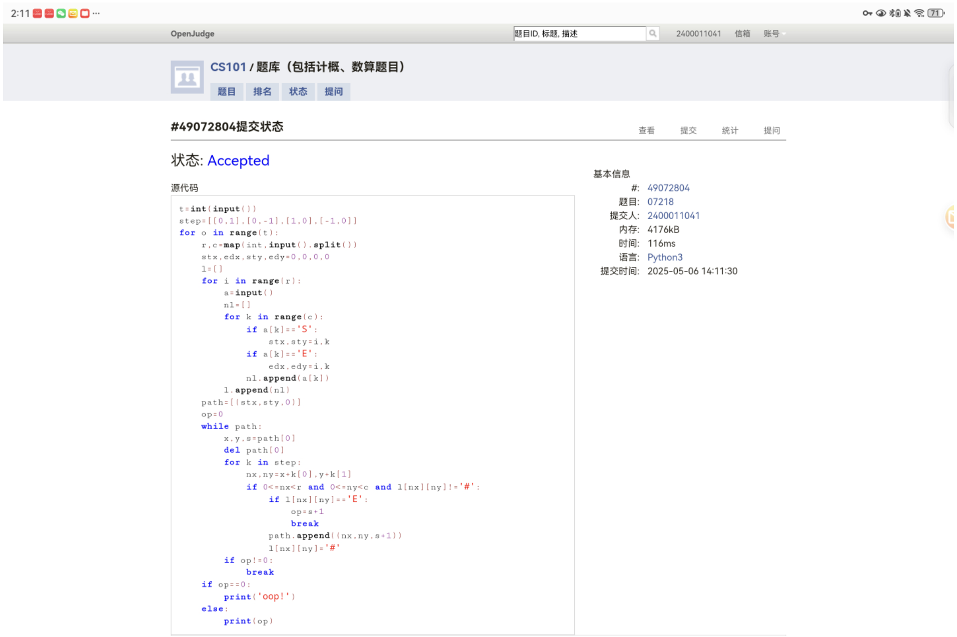
bfs, http://cs101.openjudge.cn/practice/07218/

思路：

代码：

```
t=int(input())
step=[[0,1],[0,-1],[1,0],[-1,0]]
for o in range(t):
    r,c=map(int,input().split())
    stx,edx,sty,edy=0,0,0,0
    l=[]
    for i in range(r):
        a=input()
        nl=[]
        for k in range(c):
            if a[k]=='S':
                stx,sty=i,k
            if a[k]=='E':
                edx,edy=i,k
            nl.append(a[k])
        l.append(nl)
    path=[(stx,sty,0)]
    op=0
    while path:
        x,y,s=path[0]
        del path[0]
        for k in step:
            nx,ny=x+k[0],y+k[1]
            if 0<=nx<r and 0<=ny<c and l[nx][ny]!='#':
                if l[nx][ny]=='E':
                    op=s+1
                    break
                path.append((nx,ny,s+1))
                l[nx][ny]='#'
        if op!=0:
            break
    if op==0:
        print('oop!')
    else:
        print(op)
```

代码运行截图 （至少包含有"Accepted"）



#49072804提交状态

状态: Accepted

源代码

```
t=int(input())
step=[[0,1],[0,-1],[1,0],[-1,0]]
for o in range(t):
    r,c=map(int,input().split())
    stx,edx,sty,edy=0,0,0,0
    l=[]
    for i in range(r):
        a=input()
        nl=[]
        for k in range(c):
            if a[k]=='S':
                stx,sty=i,k
            if a[k]=='E':
                edx,edy=i,k
            nl.append(a[k])
        l.append(nl)
    path=[(stx,sty,0)]
    op=0
    while path:
        x,y,s=path[0]
        del path[0]
        for k in step:
            nx,ny=x+k[0],y+k[1]
            if 0<=nx<r and 0<=ny<c and l[nx][ny]!='#':
                if l[nx][ny]=='E':
                    op=s+1
                    break
                path.append((nx,ny,s+1))
                l[nx][ny]='#'
        if op!=0:
            break
    if op==0:
        print('oop!')
    else:
        print(op)
```

基本信息
#: 49072804
题目: 07218
提交人: 2400011041
内存: 4176kB
时间: 116ms
语言: Python3
提交时间: 2025-05-06 14:11:30

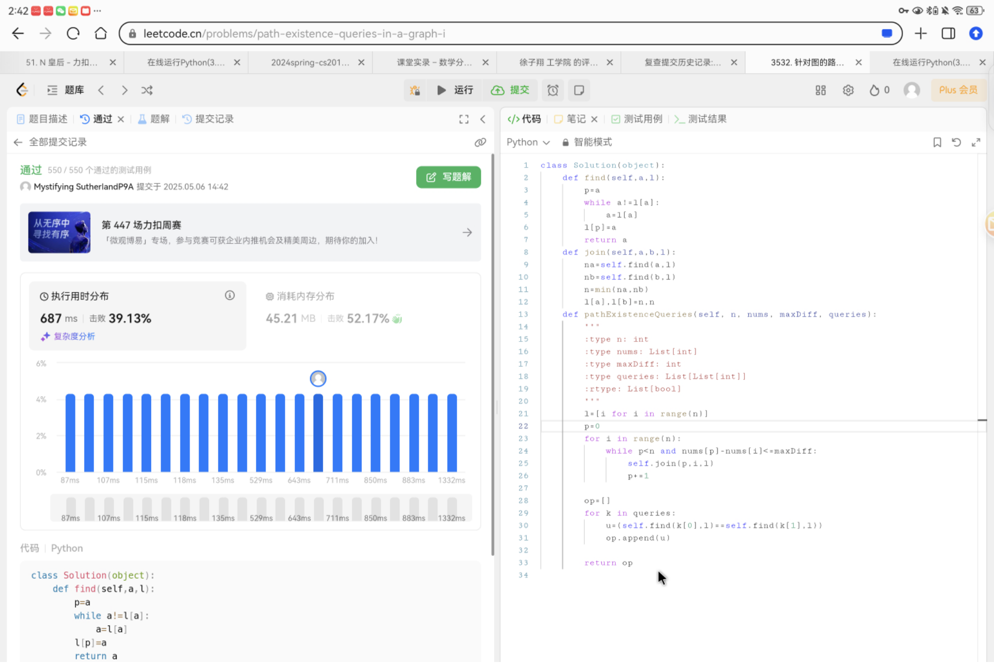
M3532.针对图的路径存在性查询 I
disjoint set, https://leetcode.cn/problems/path-existence-queries-in-a-graph-i/

思路:

代码:

```
class Solution(object):

    def find(self,a,l):

        p=a

        while a!=l[a]:

            a=l[a]

        l[p]=a

        return a

    def join(self,a,b,l):

        na=self.find(a,l)

        nb=self.find(b,l)

        n=min(na,nb)
        l[a],l[b]=n,n
    def pathExistenceQueries(self, n, nums, maxDiff, queries):
        """
        :type n: int
        :type nums: List[int]
        :type maxDiff: int
        :type queries: List[List[int]]
        :rtype: List[bool]
        """
        l=[i for i in range(n)]
        p=0
        for i in range(n):
            while p<n and nums[p]-nums[i]<=maxDiff:
                self.join(p,i,l)
                p+=1

        op=[]
        for k in queries:
            u=(self.find(k[0],l)==self.find(k[1],l))
            op.append(u)

        return op
```

代码运行截图 （至少包含有"Accepted"）



leetcode.cn/problems/path-existence-queries-in-a-graph-i

通过 550 / 550 个通过的测试用例

Mystifying SutherlandP9A 提交于 2025.05.06 14:42

第 447 场力扣周赛

「微观博易」专场，参与竞赛可获企业内推机会及精美周边，期待你的加入!

执行用时分布 687 ms | 击败 39.13%

消耗内存分布 45.21 MB | 击败 52.17%

```
class Solution(object):
    def find(self,a,l):
        p=a
        while a!=l[a]:
            a=l[a]
        l[p]=a
        return a
    def join(self,a,b,l):
        na=self.find(a,l)
        nb=self.find(b,l)
        n=min(na,nb)
        l[a],l[b]=n,n
    def pathExistenceQueries(self, n, nums, maxDiff, queries):
        """
        :type n: int
        :type nums: List[int]
        :type maxDiff: int
        :type queries: List[List[int]]
        :rtype: List[bool]
        """
        l=[i for i in range(n)]
        p=0
        for i in range(n):
            while p<n and nums[p]-nums[i]<=maxDiff:
                self.join(p,i,l)
                p+=1

        op=[]
        for k in queries:
            u=(self.find(k[0],l)==self.find(k[1],l))
            op.append(u)

        return op
```

M22528:厚道的调分方法

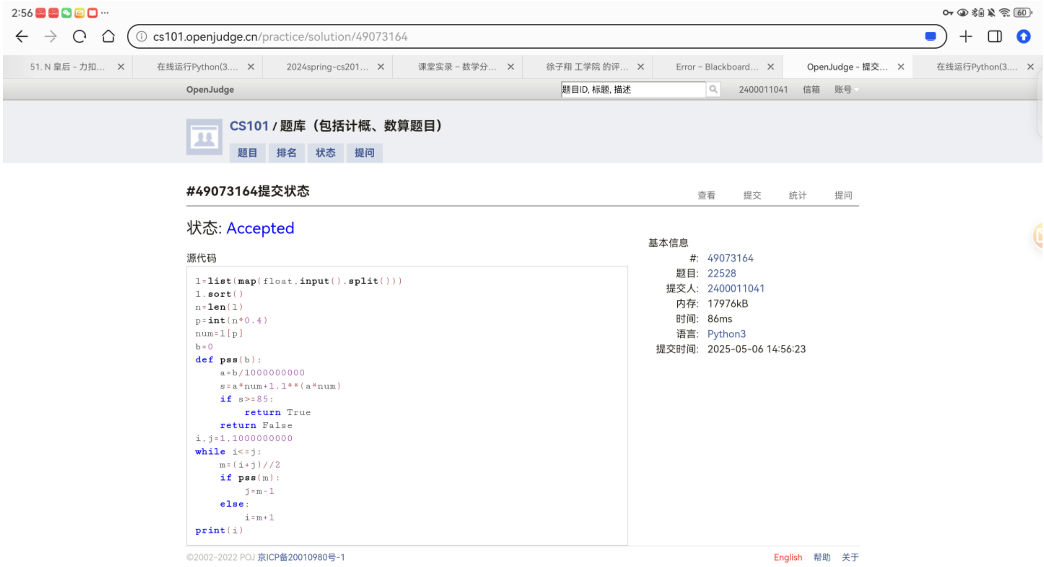
binary search, http://cs101.openjudge.cn/practice/22528/

思路：

代码：

```
l=list(map(float,input().split()))
l.sort()
n=len(l)
p=int(n*0.4)
num=l[p]
b=0
def pss(b):
    a=b/1000000000
    s=a*num+1.1**(a*num)
    if s>=85:
        return True
    return False
i,j=1,1000000000
while i<=j:
    m=(i+j)//2
    if pss(m):
        j=m-1
    else:
        i=m+1
print(i)
```

代码运行截图 （至少包含有"Accepted"）



#49073164提交状态

状态: Accepted

源代码

```
l=list(map(float,input().split()))
l.sort()
n=len(l)
p=int(n*0.4)
num=l[p]
b=0
def pss(b):
    a=b/1000000000
    s=a*num+1.1**(a*num)
    if s>=85:
        return True
    return False
i,j=1,1000000000
while i<=j:
    m=(i+j)//2
    if pss(m):
        j=m-1
    else:
        i=m+1
print(i)
```

基本信息

#:	49073164
题目:	22528
提交人:	2400011041
内存:	17976kB
时间:	86ms
语言:	Python3
提交时间:	2025-05-06 14:56:23

©2002-2022 POJ 京ICP备20010980号-1

Msy382: 有向图判环

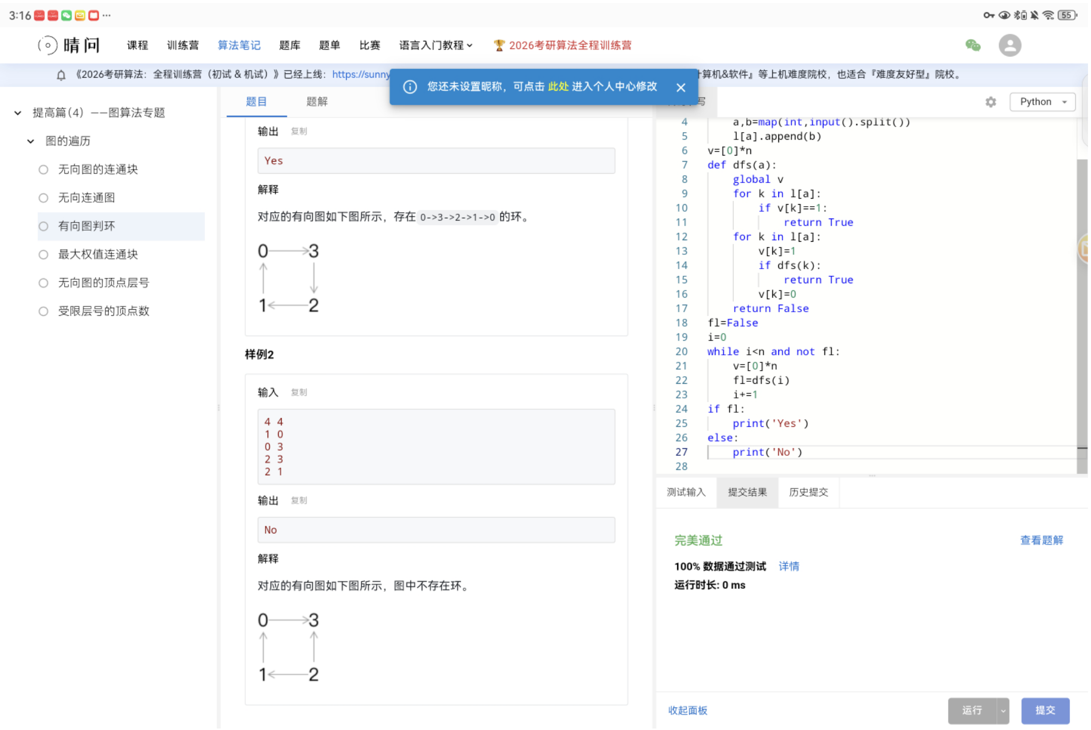
dfs, https://sunnywhy.com/sfbj/10/3/382

思路：

代码：

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

l=[[]for i in range(n)]

for i in range(m):

    a,b=map(int,input().split())

    l[a].append(b)

v=[0]*n

def dfs(a):

    global v

    for k in l[a]:

        if v[k]==1:
            return True
    for k in l[a]:
        v[k]=1
        if dfs(k):
            return True
        v[k]=0
    return False
fl=False
i=0
while i<n and not fl:
    v=[0]*n
    fl=dfs(i)
    i+=1
if fl:
    print('Yes')
else:
    print('No')
```

代码运行截图 （至少包含有"Accepted"）



输出

```
Yes
```

解释

对应的有向图如下图所示，存在`0->3->2->1->0`的环。

样例2

输入

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

输出

```
No
```

解释

对应的有向图如下图所示，图中不存在环。

```
    a,b=map(int,input().split())
    l[a].append(b)
v=[0]*n
def dfs(a):
    global v
    for k in l[a]:
        if v[k]==1:
            return True
    for k in l[a]:
        v[k]=1
        if dfs(k):
            return True
        v[k]=0
    return False
fl=False
i=0
while i<n and not fl:
    v=[0]*n
    fl=dfs(i)
    i+=1
if fl:
    print('Yes')
else:
    print('No')
```

完美通过

100% 数据通过测试

运行时长: 0 ms

M05443:兔子与樱花

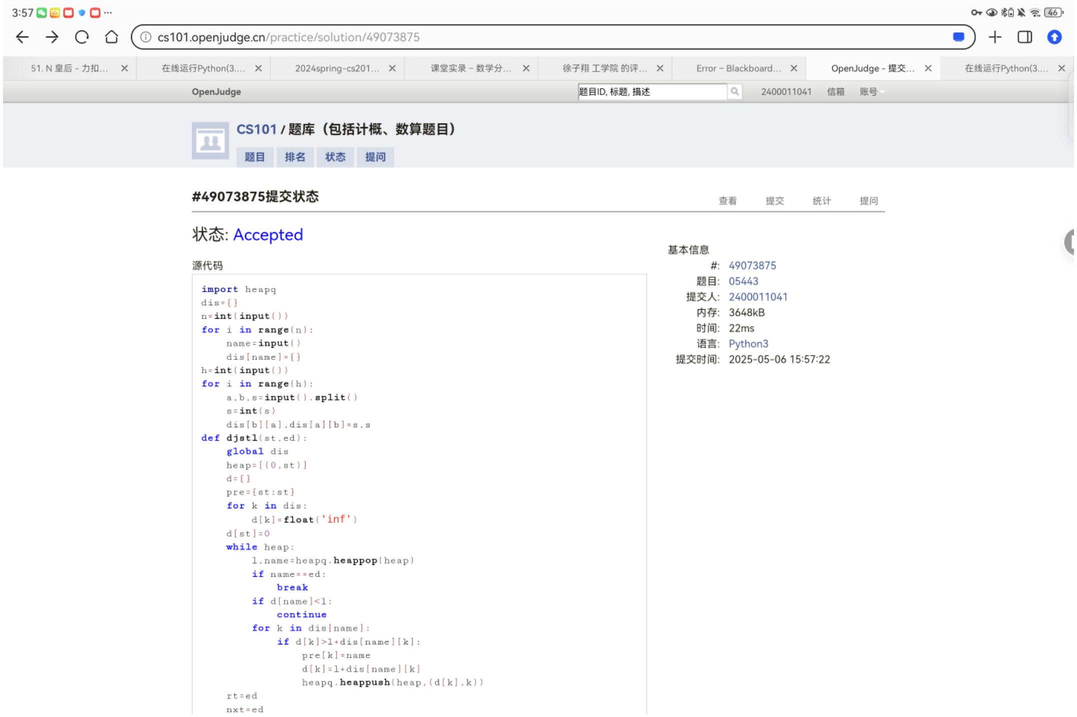
Dijkstra, http://cs101.openjudge.cn/practice/05443/

思路：

代码：

```
import heapq
dis={}
n=int(input())
for i in range(n):
    name=input()
    dis[name]={}
h=int(input())
for i in range(h):
    a,b,s=input().split()
    s=int(s)
    dis[b][a],dis[a][b]=s,s
def djstl(st,ed):
    global dis
    heap=[(0,st)]
    d={}
    pre={st:st}
    for k in dis:
        d[k]=float('inf')
    d[st]=0
    while heap:
        l,name=heapq.heappop(heap)
        if name==ed:
            break
        if d[name]<l:
            continue
        for k in dis[name]:
            if d[k]>l+dis[name][k]:
                pre[k]=name
                d[k]=l+dis[name][k]
                heapq.heappush(heap,(d[k],k))
    rt=ed
    nxt=ed
    p=pre[ed]
    while nxt!=st:
        rt=p+'->('+str(dis[p][nxt])+')->'+rt
        nxt=p
        p=pre[p]
    return rt
t=int(input())
for e in range(t):
    a,b=input().split()
    print(djstl(a,b))
```

代码运行截图　（至少包含有"Accepted"）



#49073875提交状态

状态: Accepted

基本信息

#: 49073875
题目: 05443
提交人: 2400011041
内存: 3648kB
时间: 22ms
语言: Python3
提交时间: 2025-05-06 15:57:22

源代码

```
import heapq
dis={}
n=int(input())
for i in range(n):
    name=input()
    dis[name]={}
h=int(input())
for i in range(h):
    a,b,s=input().split()
    s=int(s)
    dis[b][a],dis[a][b]=s,s
def djstl(st,ed):
    global dis
    heap=[(0,st)]
    d={}
    pre={st:st}
    for k in dis:
        d[k]=float('inf')
    d[st]=0
    while heap:
        l,name=heapq.heappop(heap)
        if name==ed:
            break
        if d[name]<l:
            continue
        for k in dis[name]:
            if d[k]>l+dis[name][k]:
                pre[k]=name
                d[k]=l+dis[name][k]
                heapq.heappush(heap,(d[k],k))
    rt=ed
    nxt=ed
```

T28050：骑士周游

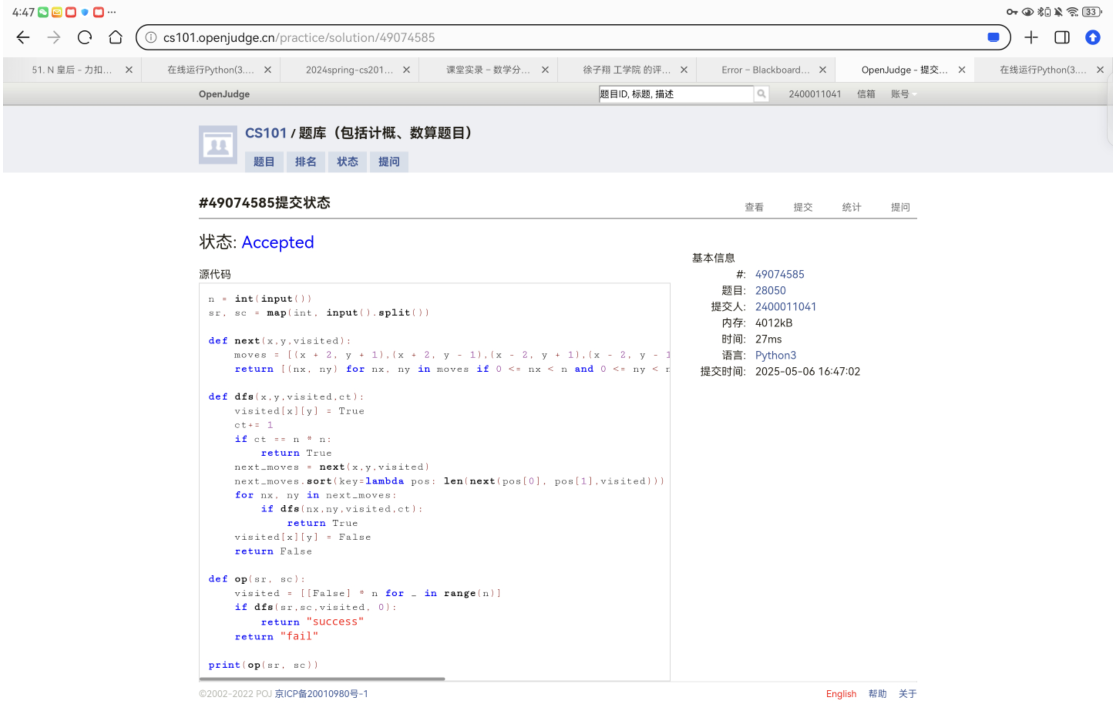
dfs, http://cs101.openjudge.cn/practice/28050/

思路:

代码:

```
n = int(input())
sr, sc = map(int, input().split())

def next(x,y,visited):
    moves = [(x + 2, y + 1),(x + 2, y - 1),(x - 2, y + 1),(x - 2, y - 1),(x + 1, y + 2), (x + 1, y - 2),(x - 1, y + 2), (x - 1, y - 2)]
    return [(nx, ny) for nx, ny in moves if 0 <= nx < n and 0 <= ny < n and not visited[nx][ny]]

def dfs(x,y,visited,ct):
    visited[x][y] = True
    ct+= 1
    if ct == n * n:
        return True
    next_moves = next(x,y,visited)
    next_moves.sort(key=lambda pos: len(next(pos[0], pos[1],visited)))
    for nx, ny in next_moves:
        if dfs(nx,ny,visited,ct):
            return True
    visited[x][y] = False
    return False

def op(sr, sc):
    visited = [[False] * n for _ in range(n)]
    if dfs(sr,sc,visited, 0):
        return "success"
    return "fail"

print(op(sr, sc))
```

代码运行截图 （至少包含有"Accepted"）



#49074585提交状态

状态: Accepted

源代码

```python
n = int(input())
sr, sc = map(int, input().split())

def next(x,y,visited):
    moves = [(x + 2, y + 1),(x + 2, y - 1),(x - 2, y + 1),(x - 2, y - 1
    return [(nx, ny) for nx, ny in moves if 0 <= nx < n and 0 <= ny < r

def dfs(x,y,visited,ct):
    visited[x][y] = True
    ct+= 1
    if ct == n * n:
        return True
    next_moves = next(x,y,visited)
    next_moves.sort(key=lambda pos: len(next(pos[0], pos[1],visited)))
    for nx, ny in next_moves:
        if dfs(nx,ny,visited,ct):
            return True
    visited[x][y] = False
    return False

def op(sr, sc):
    visited = [[False] * n for _ in range(n)]
    if dfs(sr,sc,visited, 0):
        return "success"
    return "fail"

print(op(sr, sc))
```

基本信息

#:	49074585
题目:	28050
提交人:	2400011041
内存:	4012kB
时间:	27ms
语言:	Python3
提交时间:	2025-05-06 16:47:02

©2002-2022 POJ 京ICP备20010980号-1

English 帮助 关于

2. 学习总结和收获

本次作业难度适中，就是最后一题的贪心优化难想到。借助本次作业复习了搜索相关内容。

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。

OJ 每日选做