Assignment #C: 五味杂陈

Updated 1148 GMT+8 Dec 10, 2024

2024 fall, Complied by 陈张涵、工学院系

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。

1. 题目

1115. 取石子游戏

dfs, https://www.acwing.com/problem/content/description/1117/

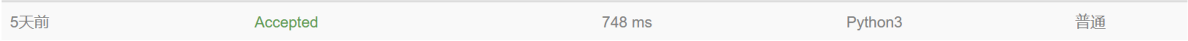
思路：根据提示非常好做

代码：

```
def who_win (a,b):
    n=1
    if a == b:
        return 'win'
    ma,mi=max(a,b),min(b,a)
    while a>0 and b>0:
        if n %2==1:
            if ma//mi>=2:
                return 'win'
            else:
                ma,mi=mi,ma-mi
        else:
            if ma//mi>=2:
                return 'lose'
            else:
                ma,mi=mi,ma-mi
        n+=1
ans=[]
while True:
    a,b=map(int,input().split())
    if a==0 and b==0:
        break
    else:
        ans.append(who_win (a,b))
```

```
for i in ans:
    print(i)
```

代码运行截图 （至少包含有"Accepted"）



5天前 Accepted 748 ms Python3 普通

25570: 洋葱

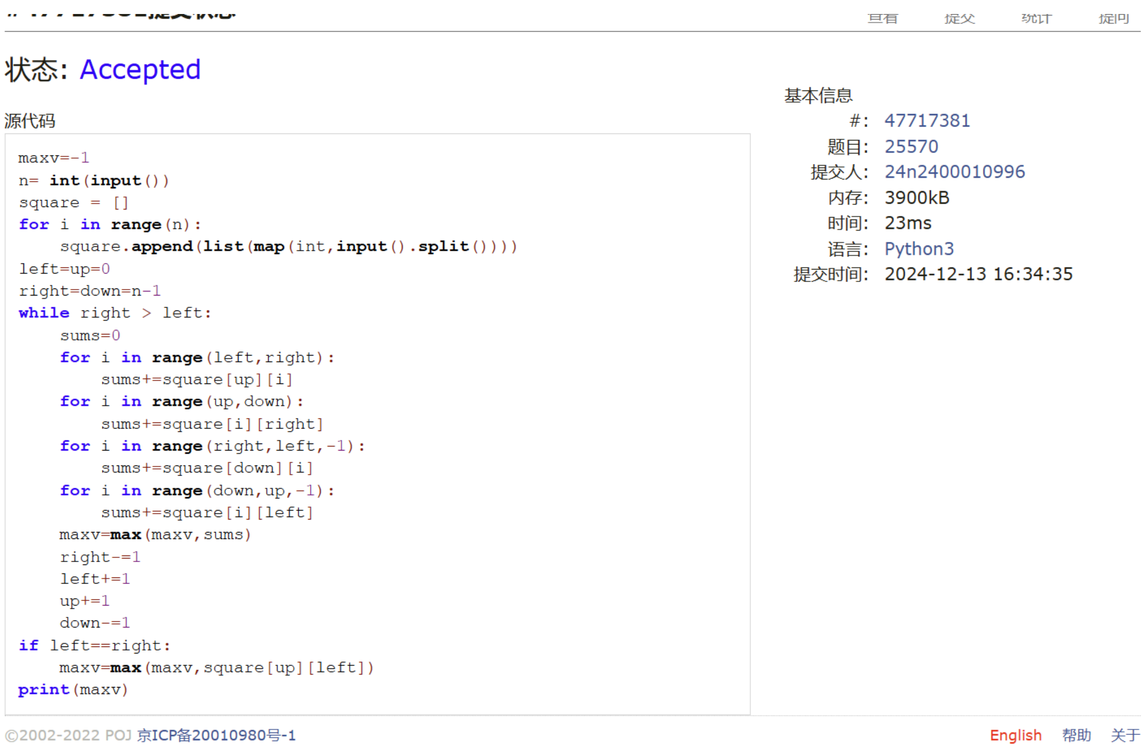
Matrices, http://cs101.openjudge.cn/practice/25570

思路：螺旋矩阵

代码：

```
maxv=-1
n= int(input())
square = []
for i in range(n):
    square.append(list(map(int,input().split())))
left=up=0
right=down=n-1
while right > left:
    sums=0
    for i in range(left,right):
        sums+=square[up][i]
    for i in range(up,down):
        sums+=square[i][right]
    for i in range(right,left,-1):
        sums+=square[down][i]
    for i in range(down,up,-1):
        sums+=square[i][left]
    maxv=max(maxv,sums)
    right-=1
    left+=1
    up+=1
    down-=1
if left==right:
    maxv=max(maxv,square[up][left])
print(maxv)
```

代码运行截图 == （至少包含有"Accepted"） ==



#47717381提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```
maxv=-1
n= int(input())
square = []
for i in range(n):
    square.append(list(map(int,input().split())))
left=up=0
right=down=n-1
while right > left:
    sums=0
    for i in range(left,right):
        sums+=square[up][i]
    for i in range(up,down):
        sums+=square[i][right]
    for i in range(right,left,-1):
        sums+=square[down][i]
    for i in range(down,up,-1):
        sums+=square[i][left]
    maxv=max(maxv,sums)
    right-=1
    left+=1
    up+=1
    down-=1
if left==right:
    maxv=max(maxv,square[up][left])
print(maxv)
```

基本信息

#:	47717381
题目:	25570
提交人:	24n2400010996
内存:	3900kB
时间:	23ms
语言:	Python3
提交时间:	2024-12-13 16:34:35

©2002-2022 POJ 京ICP备20010980号-1

English 帮助 关于

1526C1. Potions(Easy Version)

greedy, dp, data structures, brute force, *1500, https://codeforces.com/problemset/problem/1526/C1

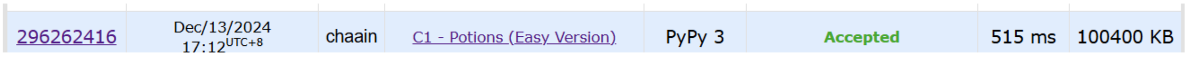
思路：很明显的dp

代码：

```
n=int(input())
drugs = list(map(int,input().split()))
dp = [[-1]*(n+1) for i in range(n+1)]
for i in range(n+1):
    dp[i][0]=0
for i in range(1,n+1):
    for j in range(1,i+1):
        if dp[i-1][j-1]>=0:
            dp[i][j]=max(dp[i-1][j],dp[i-1][j-1]+drugs[i-1])
        else:
            dp[i][j]=dp[i-1][j]

for i in range(n,-1,-1):
    if dp[n][i]>=0:
        print(i)
        break
```

代码运行截图 （至少包含有"Accepted"）



296262416	Dec/13/2024 17:12UTC+8	chaain	C1 - Potions (Easy Version)	PyPy 3	Accepted	515 ms	100400 KB

22067: 快速堆猪

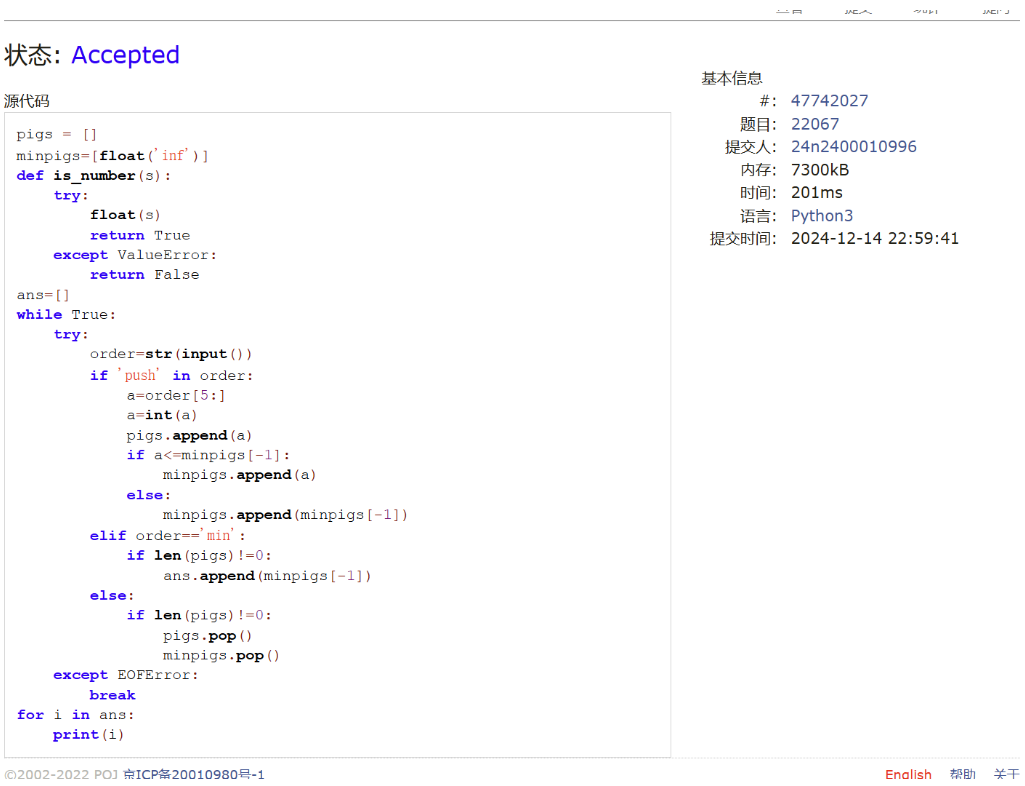
辅助栈，http://cs101.openjudge.cn/practice/22067/

思路：第一次学到辅助栈

代码：

```python
pigs = []
minpigs=[float('inf')]
def is_number(s):
    try:
        float(s)
        return True
    except ValueError:
        return False
ans=[]
while True:
    try:
        order=str(input())
        if 'push' in order:
            a=order[5:]
            a=int(a)
            pigs.append(a)
            if a<=minpigs[-1]:
                minpigs.append(a)
            else:
                minpigs.append(minpigs[-1])
        elif order=='min':
            if len(pigs)!=0:
                ans.append(minpigs[-1])
        else:
            if len(pigs)!=0:
                pigs.pop()
                minpigs.pop()
    except EOFError:
        break
for i in ans:
    print(i)
```

代码运行截图 （至少包含有"Accepted"）



状态: Accepted

源代码

```
pigs = []
minpigs=[float('inf')]
def is_number(s):
    try:
        float(s)
        return True
    except ValueError:
        return False
ans=[]
while True:
    try:
        order=str(input())
        if 'push' in order:
            a=order[5:]
            a=int(a)
            pigs.append(a)
            if a<=minpigs[-1]:
                minpigs.append(a)
            else:
                minpigs.append(minpigs[-1])
        elif order=='min':
            if len(pigs)!=0:
                ans.append(minpigs[-1])
        else:
            if len(pigs)!=0:
                pigs.pop()
                minpigs.pop()
    except EOFError:
        break
for i in ans:
    print(i)
```

基本信息

#:	47742027
题目:	22067
提交人:	24n2400010996
内存:	7300kB
时间:	201ms
语言:	Python3
提交时间:	2024-12-14 22:59:41

©2002-2022 POJ 京ICP备20010980号-1

English 帮助 关于

20106: 走山路

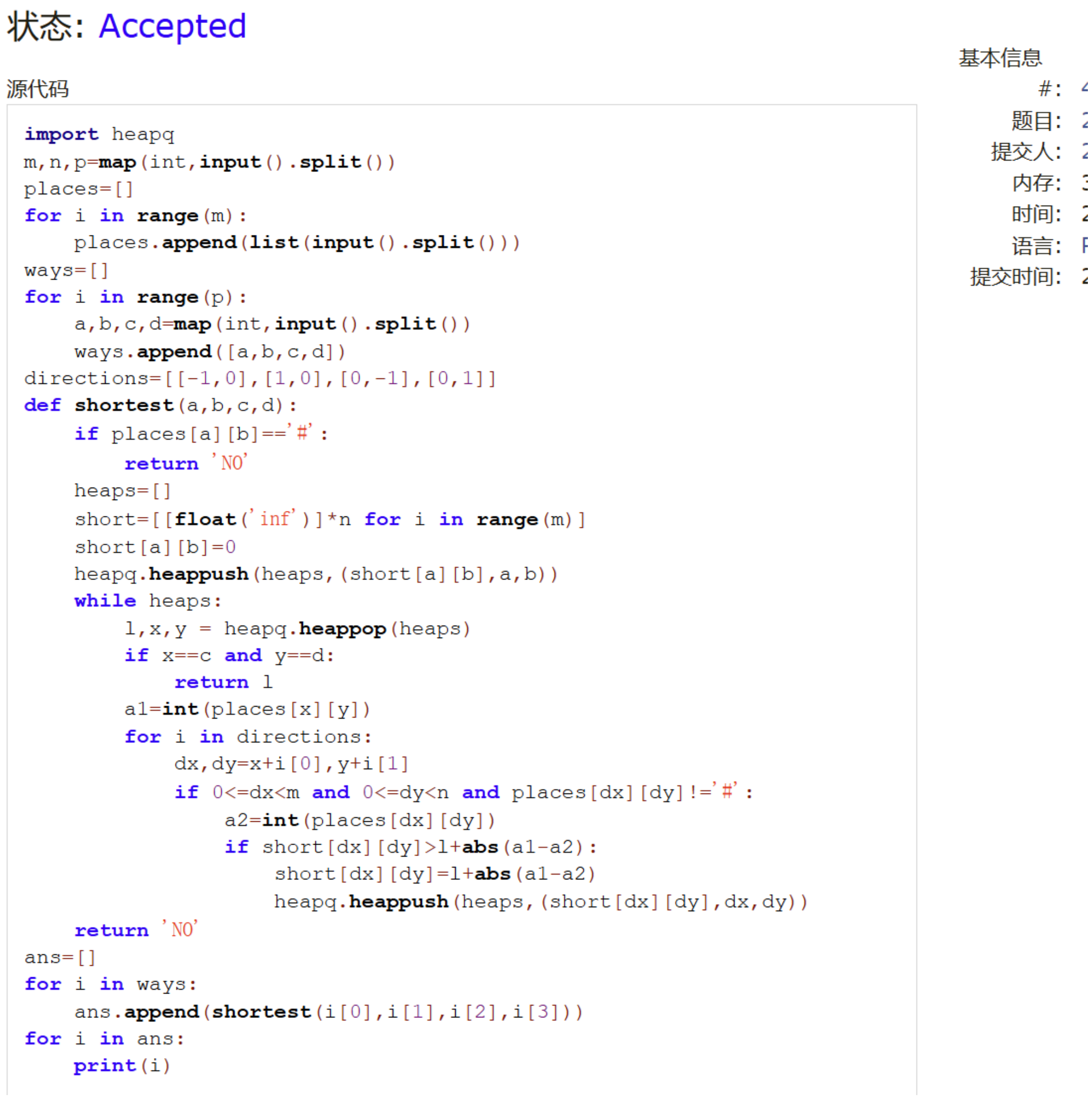
Dijkstra, http://cs101.openjudge.cn/practice/20106/

思路：看答案学了迪杰斯特拉的模版

代码：

```
import heapq
m,n,p=map(int,input().split())
places=[]
for i in range(m):
    places.append(list(input().split()))
ways=[]
for i in range(p):
    a,b,c,d=map(int,input().split())
    ways.append([a,b,c,d])
directions=[[-1,0],[1,0],[0,-1],[0,1]]
def shortest(a,b,c,d):
    if places[a][b]=='#':
        return 'NO'
    heaps=[]
    short=[[float('inf')]*n for i in range(m)]
    short[a][b]=0
    heapq.heappush(heaps,(short[a][b],a,b))
    while heaps:
        l,x,y = heapq.heappop(heaps)
        if x==c and y==d:
            return l
        a1=int(places[x][y])
        for i in directions:
            dx,dy=x+i[0],y+i[1]
            if 0<=dx<m and 0<=dy<n and places[dx][dy]!='#':
                a2=int(places[dx][dy])
                if short[dx][dy]>l+abs(a1-a2):
                    short[dx][dy]=l+abs(a1-a2)
                    heapq.heappush(heaps,(short[dx][dy],dx,dy))
    return 'NO'
ans=[]
for i in ways:
    ans.append(shortest(i[0],i[1],i[2],i[3]))
for i in ans:
    print(i)
```

代码运行截图 （至少包含有"Accepted"）



状态: Accepted

源代码

```
import heapq
m,n,p=map(int,input().split())
places=[]
for i in range(m):
    places.append(list(input().split()))
ways=[]
for i in range(p):
    a,b,c,d=map(int,input().split())
    ways.append([a,b,c,d])
directions=[[-1,0],[1,0],[0,-1],[0,1]]
def shortest(a,b,c,d):
    if places[a][b]=='#':
        return 'NO'
    heaps=[]
    short=[[float('inf')]*n for i in range(m)]
    short[a][b]=0
    heapq.heappush(heaps,(short[a][b],a,b))
    while heaps:
        l,x,y = heapq.heappop(heaps)
        if x==c and y==d:
            return l
        a1=int(places[x][y])
        for i in directions:
            dx,dy=x+i[0],y+i[1]
            if 0<=dx<m and 0<=dy<n and places[dx][dy]!='#':
                a2=int(places[dx][dy])
                if short[dx][dy]>l+abs(a1-a2):
                    short[dx][dy]=l+abs(a1-a2)
                    heapq.heappush(heaps,(short[dx][dy],dx,dy))
    return 'NO'
ans=[]
for i in ways:
    ans.append(shortest(i[0],i[1],i[2],i[3]))
for i in ans:
    print(i)
```

04129: 变换的迷宫

bfs, http://cs101.openjudge.cn/practice/04129/

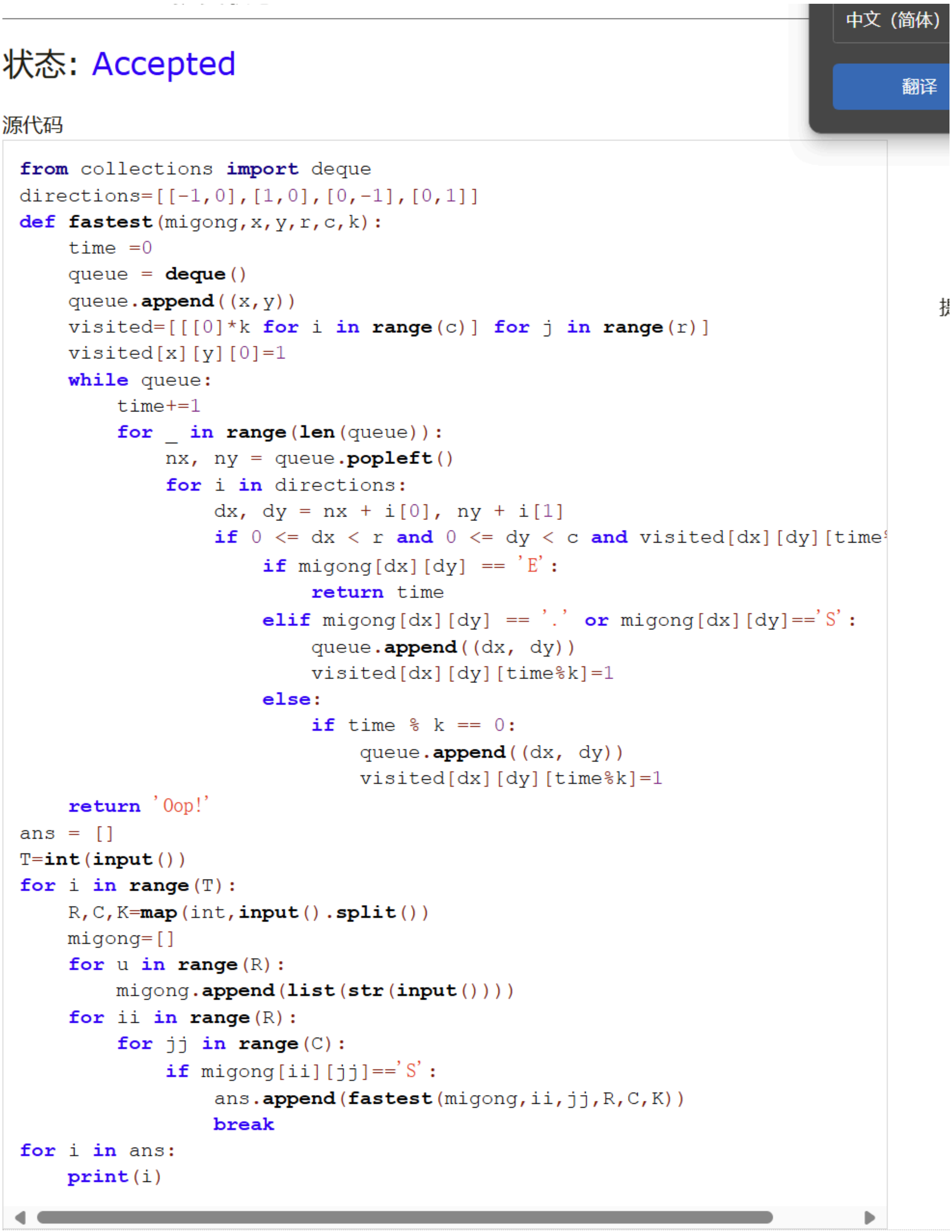
思路：

三维空间难想

代码：

```
from collections import deque
directions=[[-1,0],[1,0],[0,-1],[0,1]]
def fastest(migong,x,y,r,c,k):
    time =0
    queue = deque()
    queue.append((x,y))
    visited=set()
    visited.add((x,y,0))
    while queue:
        time+=1
        for _ in range(len(queue)):
            nx, ny = queue.popleft()
            for i in directions:
                dx, dy = nx + i[0], ny + i[1]
                if 0 <= dx < r and 0 <= dy < c and (dx,dy,time%k) not in visited:
                    if migong[dx][dy] == 'E':
                        return time
                    elif migong[dx][dy] == '.' or migong[dx][dy]=='S':
                        queue.append((dx, dy))
                        visited.add((dx,dy,time%k))
                    else:
                        if time % k == 0:
                            queue.append((dx, dy))
                            visited.add((dx,dy,time%k))
    return 'Oop!'
ans = []
T=int(input())
for _ in range(T):
    R,C,K=map(int,input().split())
    migong=[]
    for u in range(R):
        migong.append(list(str(input())))
    for j in range(R):
        for l in range(C):
            if migong[j][l]=='S':
                ans.append(fastest(migong,j,l,R,C,K))
                break
        break
for i in ans:
    print(i)
```

代码运行截图 （至少包含有"Accepted"）



状态: Accepted

源代码

```
from collections import deque
directions=[[-1,0],[1,0],[0,-1],[0,1]]
def fastest(migong,x,y,r,c,k):
    time =0
    queue = deque()
    queue.append((x,y))
    visited=[[[0]*k for i in range(c)] for j in range(r)]
    visited[x][y][0]=1
    while queue:
        time+=1
        for _ in range(len(queue)):
            nx, ny = queue.popleft()
            for i in directions:
                dx, dy = nx + i[0], ny + i[1]
                if 0 <= dx < r and 0 <= dy < c and visited[dx][dy][time
                    if migong[dx][dy] == 'E':
                        return time
                    elif migong[dx][dy] == '.' or migong[dx][dy]=='S':
                        queue.append((dx, dy))
                        visited[dx][dy][time%k]=1
                    else:
                        if time % k == 0:
                            queue.append((dx, dy))
                            visited[dx][dy][time%k]=1
    return 'Oop!'
ans = []
T=int(input())
for i in range(T):
    R,C,K=map(int,input().split())
    migong=[]
    for u in range(R):
        migong.append(list(str(input())))
    for ii in range(R):
        for jj in range(C):
            if migong[ii][jj]=='S':
                ans.append(fastest(migong,ii,jj,R,C,K))
                break
for i in ans:
    print(i)
```

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“计概2024fall每日选做”、CF、LeetCode、洛谷等网站题目。

老问题，代码一长就出bug调试好久。

感觉如果期末是这个难度或者再简单一点还是能ac4到ac5的

tough感觉自己是不太有希望了，尤其是很难的greedy或者dp

现在开始复习整理题目了