# 復習済み

A-E

# A Happy New Year 2025

正解

# B 9x9 Sum

正解

# C Snake Numbers

正解

ユーザー解説は何言ってるかわからん

# D Snaky Walk

不正解

３次元のダイクストラ法で解けるが(不正解コード)計算量のオーダーがきつい。そのためキュー用いてbfs解く。２つの状態があり、グリッドはHWマスあるので2HWマスの最短経路問題になる。ダイクストラだと だが、bfsならで解ける。

(ポイント)

最短経路の時

・辺の重みが1の普通のグリッドならbfs

・辺の重みが異なったり、グラフ構造がグリッドではないならダイクストラ

## 成功コード

from collections import deque

H, W = map(int, input().split())

S = []

INF = H\*W

for \_ in range(H):

S.append(list(input()))

for i in range(H):

for j in range(W):

if S[i][j] == 'S':

sy = i

sx = j

elif S[i][j] == 'G':

gy = i

gx = j

check = [[[INF]\*W for \_ in range(H)] for \_ in range(2)]

check[0][sy][sx] = 0

check[1][sy][sx] = 0

q = deque([(sy, sx, 0, 0), (sy, sx, 1, 0)])

while q:

ny, nx, nxt\_d, moves = q.popleft()

if nxt\_d == 0: #今から横

for dx in (1, -1):

if 0<=nx+dx<W:

if ny == gy and nx+dx == gx:

print(moves+1)

exit()

elif S[ny][nx+dx] == "." and check[nxt\_d][ny][nx+dx] > moves+1:

check[nxt\_d][ny][nx+dx] = moves+1

q.append((ny, nx+dx, 1, moves+1))

else: #今から縦

for dy in (1, -1):

if 0<=ny+dy<H:

if ny+dy == gy and nx == gx:

print(moves+1)

exit()

elif S[ny+dy][nx] == "." and check[nxt\_d][ny+dy][nx] > moves+1:

check[nxt\_d][ny+dy][nx] = moves+1

q.append((ny+dy, nx, 0, moves+1))

print(-1)

## 不正解コード(TLE)

from heapq import heapify, heappop, heappush

H, W = map(int, input().split())

S = []

for i in range(H):

s = list(input())

for j in range(W):

if s[j] == "S":

si = i

sj = j

elif s[j] == "G":

gi = i

gj = j

S.append(s)

D = [[[None]\*W for \_ in range(H)] for \_ in range(2)]

D[0][si][sj] = 0

D[1][si][sj] = 0

q = [(0, si, sj, 0), (0, si, sj, 1)]

heapify(q)

while q:

cost, i, j, state = heappop(q)

nc = cost + 1

if state == 0:

for dj in [-1, 1]:

nj = j+dj

if gi == i and gj == nj:

print(nc)

exit()

if 0<=nj<W and D[1][i][nj] == None and S[i][nj] == ".":

D[1][i][nj] = nc

heappush(q, (nc, i, nj, 1))

else:

for di in [-1, 1]:

ni = i+di

if gi == ni and gj == j:

print(nc)

exit()

if 0<=ni<H and D[0][ni][j] == None and S[ni][j] == ".":

D[0][ni][j] = nc

heappush(q, (nc, ni, j, 0))

print(-1)

# E Digit Sum Divisible 2

不正解

解説を読んで自分で書いた。双子の良い整数の性質を持つaの条件は様々存在する。そのうち一つ選ぶとする。今回は ”aの各桁の和が8でaが8の倍数であるとき、a+1も良い整数”　という条件を選んだ。このような条件はNが一定以上の時、必ず存在し意図的に作ることができる。N以下は全探索する。

## 正解コード

def sum\_each\_digit(x):

q = x

rst = 0

while q > 0:

q, r = divmod(q, 10)

rst += r

return rst

import sys

sys.set\_int\_max\_str\_digits(10\*\*7)

N = input()

if int(N) < 10\*\*4:

N = int(N)

upper = N\*2

a = N

while a < upper:

if a % sum\_each\_digit(a) == 0:

a += 1

if a % sum\_each\_digit(a) == 0:

print(a-1)

exit()

else:

a += 1

else:

a += 1

print(-1)

else:

digit = len(N)-2

head = int(N[:2])

head\_twice = head\*2

for i in range(head+1, head\_twice+1):

if sum\_each\_digit(i) == 8:

print(i\*10\*\*digit)

exit()

# F Count Arrays

解いてない

# G Prime Circuit

解いてない