# 復習済み

A-E

# A ABC400 Party

正解

# B Sum of Geometric Series

正解

# C 2^a b^2

正解

まずaは1から60くらいまでの範囲だが、3以上はaが1と2の場合ですでに列挙される。そのためaが1と2の場合だけ計算すればいいことはわかった。別解としてaを制限するのではなく、bを奇数に制限することで重複は避けられる。

# D Takahashi the Wall Breaker

不正解

<https://atcoder.jp/contests/abc400/submissions/64588063>

上記のコードは誰かの回答をただコピペして提出しただけだが、なぜこれと自分の不正解コードが違うのかよくわからなかった。最終的にヒープのデータが(a, b, c)という形式ではなく(a, (b, c))にすると高速になって、TLE(2200ms以上)から1200ms弱になった。

<https://atcoder.jp/contests/abc400/submissions/64621343>

結局その原因がAIやサイトを調べてもわからなかった。

## 不正解コード

from heapq import heapify, heappop, heappush

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

S = [input() for \_ in range(H)]

A, B, C, D = map(lambda x: int(x)-1, input().split())

d = [[float("inf")]\*W for \_ in range(H)]

d[A][B] = 0

q = [(0, A, B)]

heapify(q)

while q:

cost, ny, nx = heappop(q)

if d[ny][nx] != cost:

continue

for dy, dx in [(-1, 0), (1, 0), (0, -1), (0, 1)]:

nxtx = nx+dx

nxty = ny+dy

if 0<=nxtx<W and 0<=nxty<H:

if S[nxty][nxtx] == "#":

if d[nxty][nxtx] > cost+1:

heappush(q, (cost+1, nxty, nxtx))

d[nxty][nxtx] = cost+1

nxty\_2 = nxty + dy

nxtx\_2 = nxtx + dx

if 0<=nxtx\_2<W and 0<=nxty\_2<H and S[nxty\_2][nxtx\_2] == "#" and d[nxty\_2][nxtx\_2] > cost+1:

d[nxty\_2][nxtx\_2] = cost+1

heappush(q, (cost+1, nxty\_2, nxtx\_2))

else:

if d[nxty][nxtx] > cost:

heappush(q, (cost, nxty, nxtx))

d[nxty][nxtx] = cost

print(d[C][D])

## 正解コード

from heapq import heapify, heappop, heappush

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

S = [input() for \_ in range(H)]

A, B, C, D = map(lambda x: int(x)-1, input().split())

d = [[float("inf")]\*W for \_ in range(H)]

d[A][B] = 0

q = [(0, (A, B))]

heapify(q)

while q:

cost, tmp = heappop(q)

ny, nx = tmp

if d[ny][nx] != cost:

continue

for dy, dx in [(-1, 0), (1, 0), (0, -1), (0, 1)]:

nxtx = nx+dx

nxty = ny+dy

if 0<=nxtx<W and 0<=nxty<H:

if S[nxty][nxtx] == "#":

if d[nxty][nxtx] > cost+1:

heappush(q, (cost+1, (nxty, nxtx)))

d[nxty][nxtx] = cost+1

nxty\_2 = nxty + dy

nxtx\_2 = nxtx + dx

if 0<=nxtx\_2<W and 0<=nxty\_2<H and S[nxty\_2][nxtx\_2] == "#" and d[nxty\_2][nxtx\_2] > cost+1:

d[nxty\_2][nxtx\_2] = cost+1

heappush(q, (cost+1, (nxty\_2, nxtx\_2)))

else:

if d[nxty][nxtx] > cost:

heappush(q, (cost, (nxty, nxtx)))

d[nxty][nxtx] = cost

print(d[C][D])

# E Ringo's Favorite Numbers 3

わからなかった

以下のコードは解説を見てから書いたコード。400numberを列挙して二分探索を行うことはわかったが400numberの列挙に時間が掛かってしまった。

## 正解コード

prime\_factor\_types = [0]\*(10\*\*6+1)

for i in range(2, 10\*\*6+1):

if prime\_factor\_types[i] == 0:

for j in range(i, 10\*\*6+1, i):

prime\_factor\_types[j] += 1

num400 = []

for i in range(6, 10\*\*6+1):

if prime\_factor\_types[i] == 2:

num400.append(i\*\*2)

count\_num400 = len(num400)

Q = int(input())

for i in range(Q):

a = int(input())

ok = 0

ng = count\_num400

while ok + 1 < ng:

mid = (ok+ng)//2

if num400[mid] <= a:

ok = mid

else:

ng = mid

print(num400[ok])

# F Happy Birthday! 3

解いてない

# G Patisserie ABC 3

解いてない