# 復習済み

A-E

# A Candy Button

正解

## 正解コード

n, c = map(int, input().split())

t = list(map(int, input().split()))

ans = 1

pre = t[0]

for i in range(n-1):

if t[i+1]-pre >= c:

ans += 1

pre = t[i+1]

print(ans)

# B Hands on Ring (Easy)

正解だが、無駄があるコードになった。

ポイントは始点と終点は場合分けせず、大きい方と小さい方に分ければ場合の数は減るということ。

## 自分のコード

n, q = map(int, input().split())

left = 1

right = 2

ans = 0

for i in range(q):

h, t = input().split()

t = int(t)

if h == "R":

"右回転"

flag = 0

count = 0

while (right+count-1)%n+1 != left:

if (right+count-1)%n+1 == t:

ans += count

flag = 1

break

count += 1

if flag == 1:

right = t

continue

"左回転"

count = 1

while (right-count-1)%n+1 != left:

if (right-count-1)%n+1 == t:

ans += count

break

count += 1

right = t

elif h == "L":

"右回転"

flag = 0

count = 0

while (left+count-1)%n+1 != right:

if (left+count-1)%n+1 == t:

ans += count

flag = 1

break

count += 1

if flag == 1:

left = t

continue

"左回転"

count = 1

while (left-count-1)%n+1 != right:

if (left-count-1)%n+1 == t:

ans += count

break

count += 1

left = t

print(ans)

## 解答からヒントを得たコード

def num\_move(n, from\_, to, ng):

if from\_ > to:

from\_, to = to, from\_

if from\_ < ng < to:

return n + from\_ - to # n – (to-from)

else:

return to - from\_

n, q = map(int, input().split())

l, r = 1, 2

ans = 0

for \_ in range(q):

h, t = input().split()

t = int(t)

if h == 'L':

ans += num\_move(n, l, t, r)

l = t

else:

ans += num\_move(n, r, t, l)

r = t

print(ans)

# C Prepare Another Box

正解

解説に書いてある解法は無駄があるように思えてちゃんと確認していない。

## 正解コード

n = int(input())

A = list(map(int, input().split()))

B = list(map(int, input().split()))

A.sort()

B.sort()

flag = False

a = -1

b = -1

while a >= -(n-1):

if A[a] > B[b]:

if flag == True:

print(-1)

exit()

else:

flag = True

x = A[a]

a -= 1

continue

a -= 1

b -= 1

if b == -n:

x = A[0]

print(x)

else:

if A[0] > B[0]:

print(-1)

else:

print(x)

## 正解コード(復習)

N = int(input())

A = list(map(int, input().split()))

B = list(map(int, input().split()))

A.sort(reverse = True)

B.sort(reverse = True)

i = 0

j = 0

x = None

while i < N-1:

if A[i] <= B[j]:

i += 1

j += 1

continue

if x != None:

print(-1)

exit()

x = A[i]

i += 1

if x == None:

print(A[-1])

else:

if A[-1] <= B[-1]:

print(x)

else:

print(-1)

# D Cycle

正解

本番が1000msくらいで、復習が230msくらいなのはdequeの差。実際dequeに変更すると同じくらいのパフォーマンスが出る。

## 正解コード 1029ms

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

matrix = [[] for \_ in range(n)]

l = [-1]\*n

for i in range(m):

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

matrix[a-1].append(b-1)

q = [0]

l[0] = 0

while q:

con = q.pop(0)

for num in matrix[con]:

if num == 0:

print(l[con]+1)

exit()

if l[num] == -1:

l[num] = l[con]+1

q.append(num)

print(-1)

## 正解コード(復習) 237ms

from collections import deque

N, M = map(int, input().split())

adjacent = [[] for \_ in range(N)]

for \_ in range(M):

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

a -= 1

b -= 1

adjacent[a].append(b)

visit = [float("inf")]\*N

q = deque([(0,0)])

while q:

cost, node = q.popleft()

visit[node] = cost

for nxt in adjacent[node]:

if nxt == 0:

print(cost+1)

exit()

if visit[nxt] > cost+1:

q.append((cost+1, nxt))

print(-1)

# E Max × Sum

不正解

max Ai の候補はN-K個ある。それぞれの場合でsum Bi の最小値を出して N-Kの答えを出し、その中で最小なのが答え。復習したときも同じようなコードになった。

## 正解コード

from heapq import heapify, heappop, heappush

T = int(input())

for \_ in range(T):

N, K = map(int, input().split())

A = list(map(int, input().split()))

B = list(map(int, input().split()))

pairlist = sorted([[a, b] for a, b in zip(A,B)])

q = []

heapify(q)

sumb = 0

ans = 10\*\*18

for i in range(K-1):

sumb += pairlist[i][1]

heappush(q, -pairlist[i][1])

for i in range(K-1,N):

sumb += pairlist[i][1]

heappush(q, -pairlist[i][1])

ans = min(ans, pairlist[i][0]\*sumb)

sumb += heappop(q)

print(ans)

# F Hands on Ring (Hard)

不正解

理解できそう(見返す)

# G Treasure Hunting

不正解

理解できそう(見返す)