# 復習済み

A-E

# A Doors in the Center

正解

# B Full House 3

正解

# C Uniqueness

正解

# D Bonfire

不正解

どんどん増える煙の数に対して、一回の移動方向は全て同じなので、その移動方向を何かしらで管理すればいいと思った。頭では分かっていたがO(N^2)のコードが何かの間違いで通るかもと思ったが不正解だった。

大事なのは煙が動いていると見るのではなく、焚き木と人が動いたと見ること。

## 不正解コード

N, R, C = map(int, input().split())

S = input()

ans = [None]\*N

smoke = set()

smoke.add((0, 0))

for i in range(N):

c = S[i]

nxt = set([(0, 0)])

if c == "N":

for y, x in smoke:

nxt.add((y-1, x))

elif c == "W":

for y, x in smoke:

nxt.add((y, x-1))

elif c == "S":

for y, x in smoke:

nxt.add((y+1, x))

else:

for y, x in smoke:

nxt.add((y, x+1))

if (R, C) in nxt:

ans[i] = 1

else:

ans[i] = 0

smoke = nxt

print(\*ans, sep = "")

## 正解コード

N, R, C = map(int, input().split())

S = input()

ans = [None]\*N

smoke = set()

kindling = [0, 0]

human = [R, C]

for i in range(N):

c = S[i]

smoke.add(tuple(kindling))

if c == "N":

kindling[0] += 1

human[0] += 1

elif c == "W":

kindling[1] += 1

human[1] += 1

elif c == "S":

kindling[0] -= 1

human[0] -= 1

else:

kindling[1] -= 1

human[1] -= 1

if tuple(human) in smoke:

ans[i] = 1

else:

ans[i] = 0

print(\*ans, sep = "")

# E Tree Game

解いてない

この問題は、追加できる辺の候補をいかに速く求められるかが鍵になる。不正解コードでは各ノードからの距離を求めてたが、そこまでする必要はなく、ニ部グラフにできることに気づくと速くできる。

## 不正解コード　(TLE)

from collections import deque

N = int(input())

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

for \_ in range(N-1):

u, v = map(int, input().split())

u -= 1

v -= 1

adjacent[u].append(v)

adjacent[v].append(u)

hands = set()

for start in range(N-1):

for goal in range(start+1, N):

if goal in adjacent[start]:

continue

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

d = None

while q:

node, cost = q.popleft()

if node == goal:

d = cost

break

for nxt in adjacent[node]:

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

if d % 2 == 1:

hands.add((start, goal))

if len(hands)%2 == 1:

print("First", flush = True)

i, j = hands.pop()

print(i+1, j+1, flush = True)

else:

print("Second", flush = True)

while True:

oi, oj = map(int, input().split())

if oi == -1 and oj == -1:

exit()

hands.remove((oi-1, oj-1))

i, j = hands.pop()

print(i+1, j+1, flush = True)

## 正解コード

from collections import deque

N = int(input())

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

given\_edge = set()

for \_ in range(N-1):

u, v = map(int, input().split())

u -= 1

v -= 1

given\_edge.add((u, v))

adjacent[u].append(v)

adjacent[v].append(u)

assignment = [None]\*N

assignment[0] = 0

q = deque([0])

while q:

parent = q.popleft()

child\_num = (assignment[parent]+1)%2

for child in adjacent[parent]:

if assignment[child] == None:

assignment[child] = child\_num

q.append(child)

g0 = []

g1 = []

for i in range(N):

if assignment[i] == 0:

g0[0].append(i)

else:

g1[0].append(i)

hands = set()

for n0 in g0:

for n1 in g1:

hands.add((min(n0, n1), max(n0, n1)))

hands -= given\_edge

if len(hands)%2 == 1:

print("First", flush = True)

i, j = hands.pop()

print(i+1, j+1, flush = True)

else:

print("Second", flush = True)

while True:

oi, oj = map(int, input().split())

if oi == -1 and oj == -1:

exit()

hands.remove((oi-1, oj-1))

i, j = hands.pop()

print(i+1, j+1, flush = True)

# F ABCBA

解いてない

# G Not Only Tree Game

解いてない