**21.Missionaries and Cannibals Problem (BFS)**

from collections import deque

def is\_valid(state):

m\_left, c\_left, m\_right, c\_right, boat = state

if m\_left < 0 or c\_left < 0 or m\_right < 0 or c\_right < 0:

return False

if (m\_left > 0 and m\_left < c\_left) or (m\_right > 0 and m\_right < c\_right):

return False

return True

def get\_successors(state):

m\_left, c\_left, m\_right, c\_right, boat = state

moves = [(1, 0), (0, 1), (2, 0), (0, 2), (1, 1)]

successors = []

for m, c in moves:

if boat == 'left':

new\_state = (m\_left - m, c\_left - c, m\_right + m, c\_right + c, 'right')

else:

new\_state = (m\_left + m, c\_left + c, m\_right - m, c\_right - c, 'left')

if is\_valid(new\_state):

successors.append(new\_state)

return successors

def solve\_missionaries\_cannibals():

start = (3, 3, 0, 0, 'left')

goal = (0, 0, 3, 3, 'right')

queue = deque()

queue.append((start, [start]))

visited = set()

while queue:

state, path = queue.popleft()

if state in visited:

continue

visited.add(state)

if state == goal:

for step in path:

print(step)

return

for successor in get\_successors(state):

queue.append((successor, path + [successor]))

print("No solution found.")

solve\_missionaries\_cannibals()

OUTPUT:

