import numpy as np

def calculate\_payoff(player1\_action, player2\_action):

# Ödeme matrisini tanımla

payoff\_matrix = np.array([[3, 2], [1, 4]])

# İlgili hücredeki ödeme değerini döndür

return payoff\_matrix[player1\_action][player2\_action]

def update\_strategy(action, strategy, alpha):

# Stratejiyi güncelleme

strategy[action] += alpha \* (1 - strategy[action])

# Diğer stratejileri güncelleme

for i in range(len(strategy)):

if i != action:

strategy[i] -= alpha \* strategy[i]

if strategy[i] < 0:

strategy[i] = 0

return strategy

def normalize\_strategy(strategy):

# Stratejiyi normalize etme

strategy /= np.sum(strategy)

return strategy

def play\_game(iterations):

# Oyuncuların başlangıç stratejileri

player1\_strategy = np.array([0.5, 0.5])

player2\_strategy = np.array([0.5, 0.5])

for \_ in range(iterations):

# Stratejileri normalize etme

player1\_strategy = normalize\_strategy(player1\_strategy)

player2\_strategy = normalize\_strategy(player2\_strategy)

# Oyuncuların eylemlerini rastgele seçme

player1\_action = np.random.choice([0, 1], p=player1\_strategy)

player2\_action = np.random.choice([0, 1], p=player2\_strategy)

# Ödeme hesaplaması

player1\_payoff = calculate\_payoff(player1\_action, player2\_action)

player2\_payoff = calculate\_payoff(player2\_action, player1\_action)

# Stratejileri güncelleme

player1\_strategy = update\_strategy(player1\_action, player1\_strategy, player1\_payoff)

player2\_strategy = update\_strategy(player2\_action, player2\_strategy, player2\_payoff)

# Tahmini Nash dengesini yazdırma

print("Player 1 Strategy:", player1\_strategy)

print("Player 2 Strategy:", player2\_strategy)

# Oyunu oynatma

play\_game(1000)