import numpy as np

def solve\_zero\_sum\_game(payoff\_matrix):

num\_rows, num\_cols = payoff\_matrix.shape

# Sütunların toplamını bulma

col\_totals = np.sum(payoff\_matrix, axis=0)

# Oyuncu A'nın stratejisi (kazanma olasılıkları)

player\_a\_strategy = col\_totals / np.sum(col\_totals)

# Oyuncu B'nin stratejisi (kaybetme olasılıkları)

player\_b\_strategy = 1 - player\_a\_strategy

# Oyuncu A'nın beklenen kazançları

expected\_payoffs\_a = np.dot(payoff\_matrix, player\_b\_strategy)

# Oyuncu B'nin beklenen kazançları

expected\_payoffs\_b = np.dot(payoff\_matrix.T, player\_a\_strategy)

# Nash dengesindeki maksimum beklenen kazanç

max\_expected\_payoff = np.max(expected\_payoffs\_a)

# Nash denge stratejileri

nash\_equilibrium\_a = np.where(expected\_payoffs\_a == max\_expected\_payoff)[0]

nash\_equilibrium\_b = np.where(expected\_payoffs\_b == max\_expected\_payoff)[0]

return player\_a\_strategy, player\_b\_strategy, nash\_equilibrium\_a, nash\_equilibrium\_b

# Köşegen oyunun ödeme matrisi

payoff\_matrix = np.array([[2, 1], [3, 0]])

# Köşegen oyununu çözme

player\_a\_strategy, player\_b\_strategy, nash\_equilibrium\_a, nash\_equilibrium\_b = solve\_zero\_sum\_game(payoff\_matrix)

print("Oyuncu A'nın Stratejileri (Kazanma Olasılıkları):", player\_a\_strategy)

print("Oyuncu B'nin Stratejileri (Kaybetme Olasılıkları):", player\_b\_strategy)

print("Oyuncu A'nın Nash Dengesi Stratejileri:", nash\_equilibrium\_a)

print("Oyuncu B'nin Nash Dengesi Stratejileri:", nash\_equilibrium\_b)