

کردن وزیر ابیوس کنید.

def is_safe(board, row, col, n):

جدید کردن برای اینکه بینم وزیری در ردیف (row, col) قرار داده است

for i in range(col): # جدید کردن ردیف افقی (همچنانچه)

if board[row][i] == 1:

return False

جدید کردن قطر بالا بجهت

for i, j in zip(range(row-1, -1), range(col-1, -1)):

if board[i][j] == 1:

return False

جدید کردن قطر پائین بجهت

for i, j in zip(range(row, n, 1), range(col, -1, -1)):

if board[i][j] == 1:

return False

return True

def solve_n_queens_util(board, col, n):

حالات پایه: آگر تمام وزیرها قرار گرفته باشند

if col >= n:

return True

برای هر سلول در سطری فعلی

for i in range(n):

حکم کردن آیا می توان فریز را دراین سطح قرار داد

if is_safe(board, i, col, n):

قرار دادن فریز را دراین سطح

board[i][col] = 1

ادامه ب جسم صحیح سطح بعدی

if solve_n_queens_util(board, col + 1, n):

return True

اگر قرار رفتن فریز را دراین سطح پر حل ممکن نباشد آن را از صفحه حذف کنیم

board[i][col] = 0

اگر هیچ یک از سطوح های امتحان شده حل ممکن نباشد

return False

def solve_n_queens(n):

ایجاد صفحه سطح پر خالی

board = [[0 for _ in range(n)] for _ in range(n)]

حل ممکن باقی اخوانی اولیه از سطون اول

if not solve_n_queens_util(board, 0, n):

Print("There is no solution.")

return False

for i in range(n):

نتایج جواب

for j in range(n):

Print(board[i][j], end = " ")

Print()

return True

solve_n_queens(8)

نتایج ۸ بایناری حل ممکن باقی اخوانی کمتر