به نام خدا

- **بررسی کنید که در مسائل csp متغیری از نوع پیوسته داریم؟ با مثال**

در مسائل CSP (مسائل محدودیتهای ترتیبی) متغیرها به معمولا از نوع گسسته هستند.
یعنی فقط مقادیر خاص و مشخصی میتوانند بپذیرند اغلب این مقادیر مربوط به یک مجموعه محدود از اعداد صحیح یا مقادیر لیبل دارنده مشخص هستند.
از طرف دیگر، مسائلی که متغیرهای پیوسته دارند (continuous variables) عموماً در مسائل بهینه سازی محدب (مانند برنامه ریزی خطی و برنامهریزی غیرخطی) مورد استفاده قرار می گیرند. در این نوع مسائل مقادیر
توسط متغیرهای پیوسته به عنوان مختصات نقطه های در یک فضای پیوسته نشان داده میشوند.
پس در مسائل CSP معمولاً متغیرها از نوع گسسته هستند و نه پیوسته
مثل استفاده از متغیرهای پیوسته در مسائل CSP میتواند مسئله تقسیم منابع با مصرف انرژی مینیمم باشد در این مسئله، ما تعدادی منبع انرژی داریم که هر کدام نیاز به مقداری انرژی دارند هدف ما این است که میزان مصرف انرژی کلی را کاهش دهیم. در این حالت، متغیرهای مرتبط با میزان انرژی مصرفی (مثلا مقدار جریان برق) میتوانند به صورت پیوسته باشند.

- **PEAS ربات فوتبالیست**

عملکرد : گل زدن به تیم حریف و تلاش برای برد در مسابقات(کنترل توپ، دریبل ،دفاع،حمله،گل)و توجه ب قوانین برای خطا نکردن

معیار های موفقیت : تعداد پاس گل، تعداد پاس، تعداد برد،موقعیت نهایی در لیگ دیگر بازیکن ها

محیط : زمین های بازی فوتبال(سطح زمین میتواند از چمن مصنوعی یا طبیعی،آسفالت،مواد سخت یا شن یاشد)،

موانع(بازیکنان ،داوران،خط کشی های زمین)

شرایط(آب و هوا،نور،سروصدا)

سنسورها : بینایی(موقعیت توپ و بازیکنان و دروازه ها و موانع)

شنوایی(صدای داور و مربی و سوت داور و هم تیمی ها و صدای تشویق)

حسگر فاصله : تشخص فاصله از توپ،دروازه،موانع و کمک برای حرکت صحیح و بدون نقض قوانین فوتبال مانند برخورد بد با بازیکن ها

- تمرین 8 وزیر

```
import numpy as np

def is_safe(board, row, col):


  if np.any(board[row, :] == 1):
    return False

  if np.any(board[:, col] == 1):
    return False

  i = row
  j = col
  while i >= 0 and j >= 0:
    if board[i, j] == 1:
      return False
    i -= 1
    j -= 1
```

```
i = row
j = col
while i < 8 and j < 8:
  if board[i, j] == 1:
    return False
  i += 1
  j += 1

i = row
j = col
while i >= 0 and j < 8:
  if board[i, j] == 1:
    return False
  i -= 1
  j += 1


i = row
j = col
while i < 8 and j >= 0:
```

```
    if board[i, j] == 1:
      return False
    i += 1
    j -= 1

  return True

def solve_n_queens(board, row):

  if row == 8:

    return True

  for col in range(8):

    if is_safe(board, row, col):

      board[row, col] = 1


      if solve_n_queens(board, row + 1):
```

```
        return True

    board[row, col] = 0

  return False

board = np.zeros((8, 8), dtype=int)

if solve_n_queens(board, 0):
  print(board)
else:
  print("هیچ راه حلی برای مسئله ۸ وزیر پیدا نشد.")
```