در حوزه هوش مصنوعی، مسائل غیر قطعی و قابل مشاهده به شکل متنوعی مورد بررسی قرار می‌گیرند. این مسائل معمولاً شامل عدم قطعیت در داده‌ها، تصمیم‌گیری‌ها، وضعیت‌ها و پیش‌بینی‌ها هستند. به عنوان مثال:

1. تصمیم‌گیری در شرایط نامعلوم: هوش مصنوعی می‌تواند در مواقعی که داده‌ها ناقص یا مبهم هستند، تصمیم‌گیری کند. مثلاً در تشخیص بیماری‌ها، تصمیم‌گیری در مورد درمان مناسب بر اساس داده‌های موجود انجام می‌شود.

2. پیش‌بینی و تحلیل داده‌های غیر قطعی: هوش مصنوعی می‌تواند با استفاده از مدل‌ها و الگوریتم‌های خود، پیش‌بینی‌هایی در مورد رویدادها، بازارها، آب‌وهوا و غیره ارائه دهد.

3. تفسیر داده‌های ناقص: در مواردی که داده‌ها ناقص یا مبهم هستند، هوش مصنوعی می‌تواند با استفاده از تکنیک‌های مختلف، داده‌ها را تفسیر کند و نتایج معقولی ارائه دهد.

from typing import List, Tuple

from pprint import pprint

import random

def empty\_slots(board: List[List[int]]) -> List[Tuple[int, int]]:

    result = [(x//8, x%8) for x in range(64)]

    for i, row in enumerate(board):

        for j, slot in enumerate(row):

            if slot != 0:

                for k in range(8):

                    if (j, k) in result:

                        result.remove((j, k))

                    if (k, i) in result:

                        result.remove((k, i))

                for k in range(j, -1, -1):

                    if (k, i-abs(k-j)) in result:

                        result.remove((k, i-abs(k-j)))

                for k in range(j, 8):

                    if (k, i+abs(k-j)) in result:

                        result.remove((k, i+abs(k-j)))

                for k in range(i, -1, -1):

                    if (j+abs(k-i), k) in result:

                        result.remove((j+abs(k-i), k))

                for k in range(i, 8):

                    if (j-abs(k-i), k) in result:

                        result.remove((j-abs(k-i), k))

    return result

if \_\_name\_\_ == "\_\_main\_\_":

    left\_queens = 8

    board = None

    while left\_queens:

        left\_queens = 8

        board = [[0] \* 8 for \_ in range(8)]

        while True:

            empty = empty\_slots(board)

            if len(empty) == 0:

                break

            x, y = random.choice(empty)

            board[y][x] = 1

            left\_queens -= 1

    pprint(board)

