

BÀI TOÁN ĐẶT 8 QUÂN HẬU

```
1 chess = [[0 for x in range(8)] for y in range(8)]
2
3 def attack(i, j):
4     for k in range(8):
5         if chess[i][k]==1 or chess[k][j]==1:
6             return True
7     for k in range(8):
8         for l in range(8):
9             if (k+l==i+j) or (k-l==i-j):
10                if chess[k][l]==1:
11                    return True
12     return False
```

Trước tiên, ta tạo list của list, đại diện cho bàn cờ vua, tất cả vị trí sẽ có giá trị 0. Bàn cờ vua sẽ có kích thước 8x8, tuy nhiên bài toán này có thể áp dụng cho kích thước nxn, khi đó ta sẽ có n quân hậu.

Hàm attack dùng để kiểm tra xem việc đặt một quân hậu ở vị trí (i, j) có hợp lệ không, tức là không quân hậu khác “ăn”. Vòng lặp for thứ nhất sẽ kiểm tra hàng dọc và hàng ngang. Vòng lặp for thứ hai sẽ kiểm tra đường chéo.

```
14 def chess_n_queens(n):
15     if n==0:
16         return True
17     for i in range(8):
18         for j in range(8):
19             if (not(attack(i,j))) and (chess[i][j]!=1):
20                 chess[i][j] = 1
21                 if chess_n_queens(n-1) == True:
22                     return True
23                 chess[i][j] = 0
24     return False
```

Hàm chess_n_queens với tham số n truyền vào đại diện cho số quân hậu mà ta cần đặt.

Ta tiến hành duyệt từng ô bàn cờ theo thứ tự từ trái qua phải và từ trên xuống dưới. Nếu ô đang xét thỏa điều kiện if, tức ta có thể đặt quân hậu vào, ta sẽ gán giá trị ô đó bằng 1.

Sau đó, ta dùng đệ quy với tham số truyền vào $n-1$ để tiến hành đặt quân hậu tiếp theo. Tiếp tục cho đến khi số quân hậu cần đặt là 0.

Nếu trong quá trình đặt quân hậu, ta chưa đặt đủ 8 quân mà bàn cờ đã không còn vị trí nào có thể đặt được nữa, ta phải đặt quân hậu trước đó ở vị trí khác, nên ta có dòng code $chess[i][j] = 0$

```
26     chess_n_queens(8);
27
28     for i in chess:
29         print(i)
```

Chạy code, ta gọi hàm `chess_n_queens` với tham số truyền vào là 8. Nhắc lại, bài này áp dụng cho n quân hậu với $n \geq 4$. Do đó, thay vì khởi tạo cứng một bàn cờ 8×8 , và truyền vào hàm giá trị 8, ta có thể viết một hàm cho phép người dùng nhập n tùy ý thỏa điều kiện, sau đó ta khởi tạo bàn cờ và chạy thuật toán với giá trị n đó, khi đó bài làm sẽ trở nên linh động hơn.

```
[1, 0, 0, 0, 0, 0, 0, 0]
[0, 0, 0, 0, 1, 0, 0, 0]
[0, 0, 0, 0, 0, 0, 0, 1]
[0, 0, 0, 0, 0, 1, 0, 0]
[0, 0, 1, 0, 0, 0, 0, 0]
[0, 0, 0, 0, 0, 0, 1, 0]
[0, 1, 0, 0, 0, 0, 0, 0]
[0, 0, 0, 1, 0, 0, 0, 0]
Press any key to continue . . .
```