

Exp. No. 1

N-Queens Problem

Aim:

To solve the N-Queen problem where the goal is to place n queens on a $n \times n$ chess board such that no two queens attack each other.

Algorithm:

Step 1 - Start

Step 2 - Create a $n \times n$ chessboard with all cells set to 0, representing no queens placed.

Step 3 - Ensure no queen is in the same row, upper diagonal or lower diagonal for a given position.

Step 4 - Try placing a queen in each row of current column if it is safe using `isSafe()`.

Step 5 - Move to the next column if placing a queen works, else backtrack by removing queen.

Step 6 - If queen are placed in all columns return.

Step 7 - Display.

Program:

def is safe (board, row, col, n):

for i in range (col):

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

return false.

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

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

return false.

return true

def solve NQue till (board, col, n):

if col >= n:

return true

for i in range (n):

if is safe (board, i, col, n):

board [i][col] = 1

if solve NQ (board, col+1, n) == true:

return true

board [i][col] = 0

return false.

def solve NQ (n):

board = [0] * n

if solve NQ until (board, 0, n) == false:

print ("solution does not exist")

return false.

for i in board:

Output

Enter n value : 5

[1, 0, 0, 0, 0]

[0, 0, 0, 1, 0]

[0, 1, 0, 0, 0]

[0, 0, 0, 0, 1]

[0, 0, 1, 0, 0]

Result:

Thus the n-queens problem is executed & the output is verified successfully.