

Design n-Queens matrix having first Queen placed. Use backtracking to place remaining. Queens to generate the final n-queen's matrix

```
In [4]: def is_safe(board, row, col, N):
# Check this row on left side
for i in range(col):
    if board[row][i] == 1:
        return False

# Check upper diagonal on left side
for i, j in zip(range(row, -1, -1), range(col, -1, -1)):
    if board[i][j] == 1:
        return False

# Check lower diagonal on left side
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(board, col, N, first_col):
    if col >= N:
        return True

    # Skip the column where the first queen is already placed
    if col == first_col:
        return solve_n_queens(board, col + 1, N, first_col)

    for i in range(N):
        if is_safe(board, i, col, N):
            board[i][col] = 1
            if solve_n_queens(board, col + 1, N, first_col):
                return True
            board[i][col] = 0 # Backtrack

    return False

def n_queens_with_first_queen(N, first_row, first_col):
    # Initialize the board
    board = [[0 for _ in range(N)] for _ in range(N)]

    # Place the first queen
    board[first_row][first_col] = 1

    # Try to solve the rest of the board starting from the next column
    if not solve_n_queens(board, 0, N, first_col):
        print("Solution does not exist")
        return None

    return board

# Get user input
N = int(input("Enter the size of the board (N): "))
first_row = int(input(f"Enter the row (0 to {N-1}) where the first queen is placed: "))
first_col = int(input(f"Enter the column (0 to {N-1}) where the first queen is placed: "))
```

```
# Validate the input
if 0 <= first_row < N and 0 <= first_col < N:
    board = n_queens_with_first_queen(N, first_row, first_col)
    if board:
        print("Final N-Queens matrix:")
        for row in board:
            print(row)
    else:
        print("Invalid input! Please enter valid row and column values.")
```

Enter the size of the board (N): 8

Enter the row (0 to 7) where the first queen is placed: 0

Enter the column (0 to 7) where the first queen is placed: 0

Final N-Queens matrix:

[1, 0, 0, 0, 0, 0, 0, 0]

[0, 0, 0, 0, 0, 0, 1, 0]

[0, 0, 0, 0, 1, 0, 0, 0]

[0, 0, 0, 0, 0, 0, 0, 1]

[0, 1, 0, 0, 0, 0, 0, 0]

[0, 0, 0, 1, 0, 0, 0, 0]

[0, 0, 0, 0, 0, 1, 0, 0]

[0, 0, 1, 0, 0, 0, 0, 0]

In []: