N Queen Problem - backtracking

Code:

In [1]:

```
# global N
N = int(input("Enter the value of N: "))

def printSolution(board):
    for i in range(N):
        for j in range(N):
            print ('|',board[i][j],end=' ')
        print('|')

def isSafe(board, row, col):
    for i in range(col):
        if board[row][i] == 'Q':
            return False

    for i, j in zip(range(row, -1, -1), range(col, -1, -1)):
        if board[i][j] == 'Q':
            return False

    for i, j in zip(range(row, N, 1), range(col, -1, -1)):
        if board[i][j] == 'Q':
            return False

    return True

def solveNQUtil(board, col):
    if col >= N:
        return True

    for i in range(N):
        if isSafe(board, i, col):
            board[i][col] = 'Q'

            if solveNQUtil(board, col + 1) == True:
                return True

            board[i][col] = ' '

    return False

def solveNQ():
    board = [[' ']*N for _ in range(N)]

    if solveNQUtil(board, 0) == False:
        print ("Solution does not exist")
        return False

    printSolution(board)
    return True

solveNQ()

# Output:
```

```
Enter the value of N: 8
| Q |   |   |   |   |   |   |   |
|   |   |   |   |   |   | Q |   |
|   |   |   |   | Q |   |   |   |
|   |   |   |   |   |   |   | Q |
|   | Q |   |   |   |   |   |   |
|   |   |   | Q |   |   |   |   |
|   |   |   |   |   | Q |   |   |
|   |   | Q |   |   |   |   |   |
```

Out[1]:

```
True
```