N Queen Problem - branch & bound

Code:

In [ ]:

```
N = 8

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

def isSafe(row, col, slashCode, backslashCode, rowLookup, slashCodeLookup, backslashCode
    if (slashCodeLookup[slashCode[row][col]] or backslashCodeLookup[backslashCode[row][c
        return False

    return True

def solveNQueensUtil(board, col, slashCode, backslashCode, rowLookup, slashCodeLookup, b
    if(col >= N):
        return True
    for i in range(N):
        if(isSafe(i, col, slashCode, backslashCode, rowLookup, slashCodeLookup, backslas
            board[i][col] = 'Q'
            rowLookup[i] = True
            slashCodeLookup[slashCode[i][col]] = True
            backslashCodeLookup[backslashCode[i][col]] = True

            if(solveNQueensUtil(board, col + 1, slashCode, backslashCode, rowLookup, sla
                return True

            board[i][col] = ' '
            rowLookup[i] = False
            slashCodeLookup[slashCode[i][col]] = False
            backslashCodeLookup[backslashCode[i][col]] = False

    return False

def solveNQueens():
    board = [[' ' for i in range(N)]
                for j in range(N)]
    slashCode = [[' ' for i in range(N)]
                    for j in range(N)]
    backslashCode = [[' ' for i in range(N)]
                        for j in range(N)]

    rowLookup = [False] * N
    x = 2 * N - 1
    slashCodeLookup = [False] * x
    backslashCodeLookup = [False] * x

    for rr in range(N):
        for cc in range(N):
            slashCode[rr][cc] = rr + cc
            backslashCode[rr][cc] = rr - cc + 7

    if(solveNQueensUtil(board, 0, slashCode, backslashCode, rowLookup, slashCodeLookup,
        print("Solution does not exist")
        return False

    printSolution(board)
    return True
```

```
solveNQueens()

#Output
```

```
|Q| | | | | | | |
| | | | | | |Q| |
| | | | |Q| | | |
| | | | | | | |Q|
| |Q| | | | | | |
| | | |Q| | | | |
| | | | | |Q| | |
| | |Q| | | | | |
```

Out[10]:

```
True
```