```
def solveNQueens(n):
    # Initialize the sets for columns, positive diagonals, negative diagonals, list
    # for solutions
    col = set()
    posDiag = set() # (r+c)
    negDiag = set() # (r-c)
    solutions = []
    # Create the starting board of n x n matrix for "chess board"
    board = [["."]*n for i in range(n)]
    print(board)

    def successorFunction(r):
      # This is for when you have completed all rows in chess board and the row
      # index now equals n
      if r == n:
        copy = ["".join(row) for row in board]
        # Appends a copy of the solution we've reached.
        solutions.append(copy)
        print('Solution Found!')
        # Returns in order to go back up the branch to the root.
        return
      # This is iterating over 0 through 1 less than n (every column) and checking
      # if the column has already been used, if the positive diagonal has already
      # been used, or the negative diagonal has already been used. If any of them
      # have been used, it will jump to the next column value in this row.
      # If none of them have been used, it will jump to the next comment.
      for c in range(n):
        if ((c in col) or ((r+c) in posDiag) or ((r-c) in negDiag)):
          continue

        # If you jumped to this comment, then the current column values haven't
        # been assigned to any of the 3 sets and we will now add them.
        col.add(c)
        posDiag.add(r+c)
        negDiag.add(r-c)
        # After we've added these values, we will now assign Q to this column
        # in this row.
        board[r][c] = "Q"
        print('edited Board')
        print(board)

        # We will now recursively call the successor function and increment the row
        successorFunction(r+1)

        # When the previous recursive call completes, these will clear the sets of
        # values related to that specific branch.  When it hits the root, it will begir
        # the iterative process of the next branch.
        print('begin removing')
        col.remove(c)
        posDiag.remove(r+c)
        negDiag.remove(r-c)
        board[r][c] = "."

    # Initial call to the successorFunction
    successorFunction(0)
    # Returns the complete list of solutions
    return solutions

# Change the value being passed into solveNQueens in order to change the size of
# "chess board" and number of Queens
x = solveNQueens(4)
count = 0
for i in x:
  count+=1
  print(i)

print('Number of Solutions: ' + str(count))
```

```
[['.', '.', '.', '.'], ['.', '.', '.', '.'], ['.', '.', '.', '.'], ['.', '.', '.', '.']]
edited Board
[['Q', '.', '.', '.'], ['.', '.', '.', '.'], ['.', '.', '.', '.'], ['.', '.', '.', '.']]
edited Board
[['Q', '.', '.', '.'], ['.', '.', 'Q', '.'], ['.', '.', '.', '.'], ['.', '.', '.', '.']]
begin removing
edited Board
[['Q', '.', '.', '.'], ['.', '.', '.', 'Q'], ['.', '.', '.', '.'], ['.', '.', '.', '.']]
```

```
edited Board
[['Q', '.', '.', '.'], ['.', '.', '.', 'Q'], ['.', 'Q', '.', '.'], ['.', '.', '.', '.']]
begin removing
begin removing
begin removing
edited Board
[['.', 'Q', '.', '.'], ['.', '.', '.', '.'], ['.', '.', '.', '.'], ['.', '.', '.', '.']]
edited Board
[['.', 'Q', '.', '.'], ['.', '.', '.', 'Q'], ['.', '.', '.', '.'], ['.', '.', '.', '.']]
edited Board
[['.', 'Q', '.', '.'], ['.', '.', '.', 'Q'], ['Q', '.', '.', '.'], ['.', '.', '.', '.']]
edited Board
[['.', 'Q', '.', '.'], ['.', '.', '.', 'Q'], ['Q', '.', '.', '.'], ['.', '.', 'Q', '.']]
Solution Found!
begin removing
begin removing
begin removing
begin removing
edited Board
[['.', '.', 'Q', '.'], ['.', '.', '.', '.'], ['.', '.', '.', '.'], ['.', '.', '.', '.']]
edited Board
[['.', '.', 'Q', '.'], ['Q', '.', '.', '.'], ['.', '.', '.', '.'], ['.', '.', '.', '.']]
edited Board
[['.', '.', 'Q', '.'], ['Q', '.', '.', '.'], ['.', '.', '.', 'Q'], ['.', '.', '.', '.']]
edited Board
[['.', '.', 'Q', '.'], ['Q', '.', '.', '.'], ['.', '.', '.', 'Q'], ['.', 'Q', '.', '.']]
Solution Found!
begin removing
begin removing
begin removing
begin removing
edited Board
[['.', '.', '.', 'Q'], ['.', '.', '.', '.'], ['.', '.', '.', '.'], ['.', '.', '.', '.']]
edited Board
[['.', '.', '.', 'Q'], ['Q', '.', '.', '.'], ['.', '.', '.', '.'], ['.', '.', '.', '.']]
edited Board
[['.', '.', '.', 'Q'], ['Q', '.', '.', '.'], ['.', '.', 'Q', '.'], ['.', '.', '.', '.']]
begin removing
begin removing
edited Board
[['.', '.', '.', 'Q'], ['.', 'Q', '.', '.'], ['.', '.', '.', '.'], ['.', '.', '.', '.']]
begin removing
begin removing
['.Q..', '...Q', 'Q...', '..Q.']
['..Q.', 'Q...', '...Q', '.Q..']
Number of Solutions: 2
```