

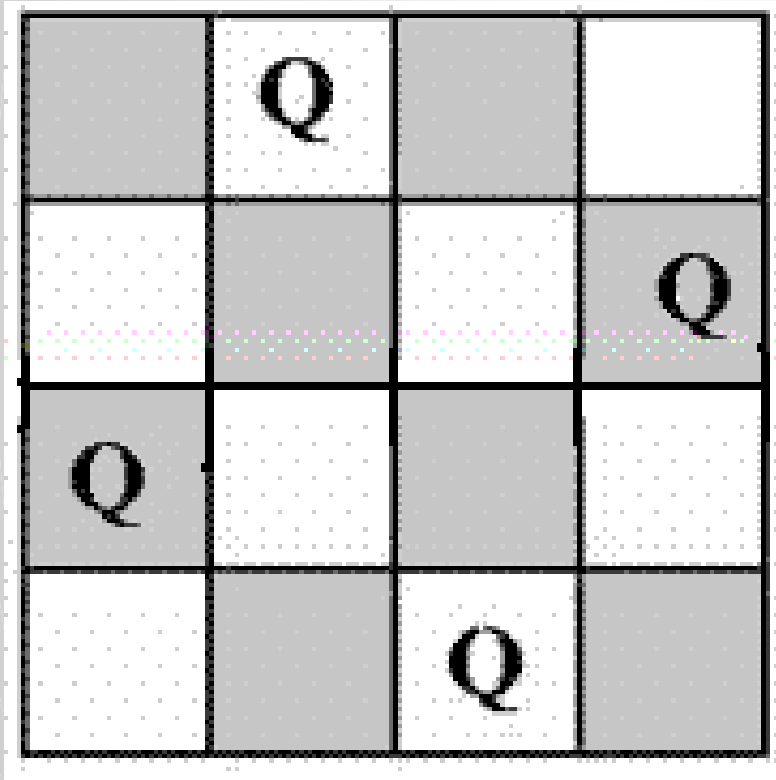
# **Solving N queen problem by backtracking**

Sagar Giri

# Introduction

- The N Queen is the problem of placing N chess queens on an  $N \times N$  chessboard so that no two queens attack each other.
- What that means is we need to find the configuration of N queens where any of the two queens do not share a row, column and diagonal paths.

**For example, following is a solution for 4 Queen problem.**



# Naive Algorithm

Generate all possible configurations of queens on board and print a configuration that satisfies the given constraints.

# Backtracking Algorithm

- The idea is to place queens one by one in different columns, starting from the leftmost column.
- When we place a queen in a column, we check for clashes with already placed queens.
- In the current column, if we find a row for which there is no clash, we mark this row and column as part of the solution.
- If we do not find such a row due to clashes then we backtrack and return false.

# Backtracking Algorithm

- 1) Start in the leftmost column
- 2) If all queens are placed  
    return true
- 3) Try all rows in the current column. Do following for every tried row.
  - a) If the queen can be placed safely in this row then mark this [row, column] as part of the solution and recursively check if placing queen here leads to a solution.
  - b) If placing queen in [row, column] leads to a solution then return true.
  - c) If placing queen doesn't lead to a solution then unmark this [row, column] (Backtrack) and go to step (a) to try other rows.
- 3) If all rows have been tried and nothing worked, return false to trigger backtracking.

# Thank You

Sources:

<http://www.geeksforgeeks.org/backtracking-set-3-n-queen-problem/>

<https://www.youtube.com/watch?v=JkP-xats3no>

<https://sadakurapati.wordpress.com/2013/12/10/n-queens-backtracking-algorithm/>