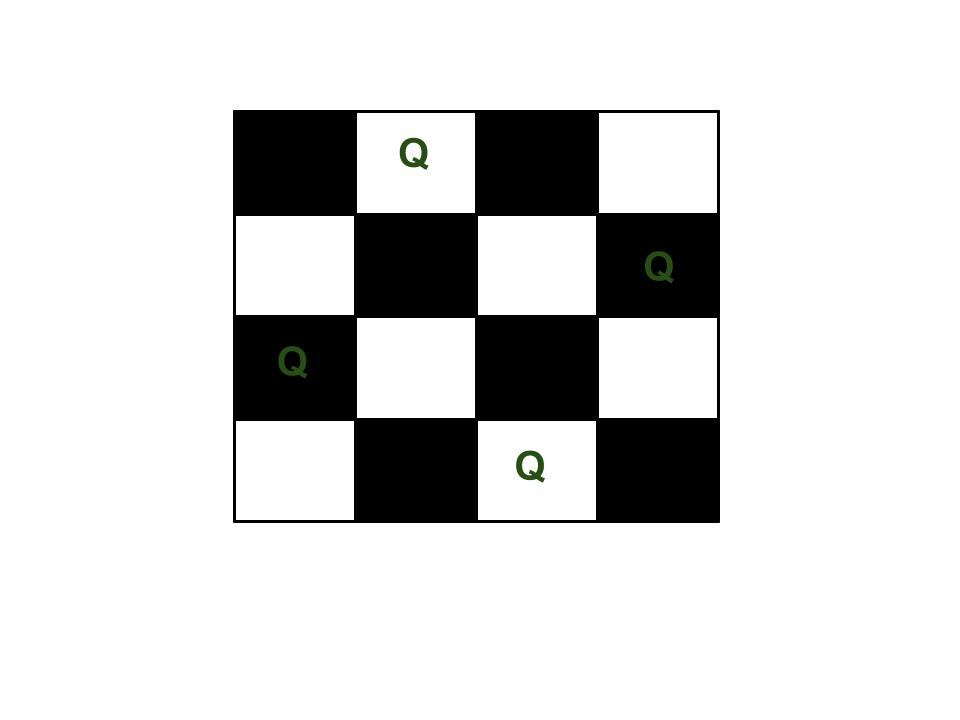
**N Queen Problem | Backtracking-3**

Let us discuss N Queen as another example problem that can be solved using backtracking.   
The N Queen is the problem of placing N chess queens on an N×N chessboard so that no two queens attack each other. For example, the following is a solution for the 4 Queen problem.

The expected output is in form of a matrix that has ‘Q’s for the blocks where queens are placed and the empty spaces  are represented by ‘.’s . For example, the following is the output matrix for the above 4 queen solution.

**Algorithm for N queen problem:-**

* Initialize an empty chessboard of size NxN.
* Start with the leftmost column and place a queen in the first row of that column.
* Move to the next column and place a queen in the first row of that column.
* Repeat step 3 until either all N queens have been placed or it is impossible to place a queen in the current column without violating the rules of the problem.
* If all N queens have been placed, print the solution.
* If it is not possible to place a queen in the current column without violating the rules of the problem, backtrack to the previous column.
* Remove the queen from the previous column and move it down one row.
* Repeat steps 4-7 until all possible configurations have been tried.

**Backtracking Algorithm** **Method 1:**  
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.

Method 1:

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 the 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.

4) If all rows have been tried and nothing worked,

return false to trigger backtracking.