IT081 Pankhania Aanandi R.

DAA -

N-Queen Problem using Backtracking

Problem Analysis:-

The problem is to find an arrangement of N queens on a chess board, such that no queen can attack any other queens on board.

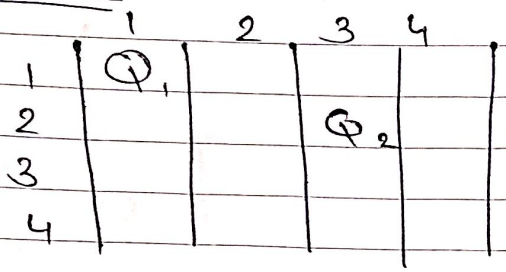
The chess queens can attack in any direction as horizontal, vertical, & diagonal way.

Binary matrix is used to display the position of N queens. where no queens can attack other queens.

[∵ NXN board]

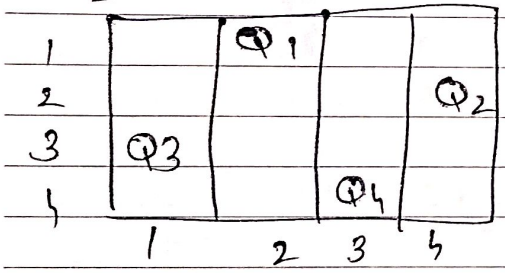
INPUT :- N = 4

step-1 :-



first column first row now for second we can't place it in 2nd colm (dia.) so placed in 3rd column. now for 3rd queen we can't place it in column 1 bcz Q1 is there not in 2nd & 4th because then it'll be diagonally placed with Q2 not in 3rd also because Q2 is placed there so,... placing Q1 in second column...

step-2 :-



Q2 → 4th (not 1, 2, 3 possible)

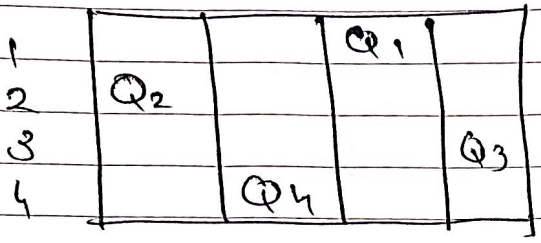
Q3 → 1st (2 → Q1, 3, 4 → Q2)

Q4 → 3rd (1, 2 → Q3, Q1, Q4 → Q2)

(Trace For : row, column, dia. rule defined)

other solution possible? yes. (mirror image)

placing Q1 in 3rd column ...



∴ (no violations are occured)
↳ row, colm, diag.

Algorithm :-

```
Algorithm place (k, i)
// return true if x is placed at
// kth row and ith column,
// return false otherwise
// x[] is a global integer array
{
    for j=1 to (k-1) do
        if ( x[j] = i ) // in same colm
            or (abs (x[j]-i) = abs (j-k)) // diag same
                return false
    return true
}
```

```
Algorithm Nqueens (k, n)
//using backtracking
{
    for i=1 to n do
        if (place (k, i)) then
            x[k] = i          // store colm in x[]
        if (k == n) then
            print (x[1:n])    // print soln
        else
            Nqueens(k+1, n)   // recursive
```

Time complexity :-

$O(n^n)$: it will trace every position on an nxn board n times for n queens.