IT081 Pankhania Aanandi R.

9 DAA -

N-Queen Problem using Backtracking

10

Problem Analysis:-

11

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

2 The chess queens can attack in
any direction as horizontal, vertical, &
3 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 :-

	1	2	3	4
1	Q_1			
2			Q_2	
3				
4				

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 :-

	1	2	3	4
1		Q1		
2				Q2
3	Q3			
4			Q4	

Q2 → 4th (not 1,2,3 possible)

Q3 → 1st (2 → Q1
3,4 → Q2)

Q4 → 3rd (1,2 → Q3, Q1
~~Q~~·4 → Q2)

(Trace for: row, column, dia. rule defined) other solution possible? yes. (mirror image)

placing Q1 in 3rd column...

1			Q_1	
2	Q_2			
3				Q_3
4		Q_4		

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

Algorithm:-

```
Algorithm place (k, i)
// return true if q 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)) // dig 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 col$^n$ in x arr
        if (k = = n) then
            print (x [1 : n])          // print sol$^n$
        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.