Roll No: 33231
Class : TE IO

CamlinExam
PAGE :
DATE :

ASSIGNMENT 1

Problem Statement : Position n-queens on a n*n chessboard such that no two queens are in the same row, columns or diagonals.

Input : Positive Integer N.
Output : All possible ways n queens can be placed on a nxn chessboard so that no two queens attack each other.

Theory :

What is Backtracking ?

Backtracking is a technique based on algorithm to solve problem. It uses recursive calling to find the solution by building a solution step by step increasing values with time. It removes the solutions that don't give rise to the solution of the problem based on the constraints given to solve the problem.

Backtracking solution for N-queens :

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 goto step (a) to try other rows.

3) If all rows have been ~~triggered~~ tried and nothing worked return false to trigger backtracking.

N-Queens Problem:

- In implementing the n-queens problem we imagine the chessboard as a two-dimensional array.
- The condition to test whether two queens at (i, j) and (k, l) are on the same row, column is simply to check ~~abs~~(l = k or j = l
- The conditions to check whether two queens are on same diagonal or not are to be found

(1,1)	1,2	1,3	1,4
2,1	2,2	2,3	2,4
3,1	3,2	3,3	3,4
4,1	4,2	4,3	4,4

i) For the elements in the upper left or lower right diagonal, the row-column values are same or

row - col = 0 Eg: 1-1 = 2-2 = 3-3 = 4-4 = 0

ii) For the elements in the upper right to the lower left-diagonal, row + col value is the same
Eg: 1+4 = 2+3 = 3+2 = 4+1 = 5

Thus, 2 queens are placed at (i,j), (k,l) then they are on same diagonal only if $i-j = k-l$ or $i+j = k+l$

2 queens lie on the same diagon if and only if $|j-l| = |i-k|$

Brute Force Approach to solve N-queens :
Generate all possible configrations of queens on board and print a configuration that satisfies the given constraints

```
while there are untried configs {
    generate the next config
    if queens do not attack in this config {
        print this config
    }
}
```

Pseudocode for N-queens problem :

```
1) Algorithm NQueens(k, n)
2) // using backtracking, this procedure prints all
   possible placements of n queens on a n*n chess-
```

// board so that they are not attacking each other

3) for i := 1 to n do {
    if place (k, i) then {
        x[k] := i;
        if (k = n) then write (x[1:n]);
        else N queens (k+1, n);
    }
}

1) Algorithm Place (k, i)
2) // return true if queen can be placed in k^{th} row and
// i^{th} column. Otherwise return false. x[] is a global
// array whose first (k-1) value has been set abs(r)
// returns absolute value of r
3) for j := 1 to k-1 do
    if ((x[j] = i) // two in same col
    or (abs(x[j] - i) = abs(j - k)))
        // or in same diagonal
        then return ~~true~~ false;
    return true;
}

Conclusion :

I have understood ~~and~~ ~~suttle~~ the concept of Backtracking and successfully implemented N-queens problem.