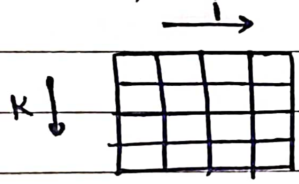
* N queens problem

place n queens in such a manner on an $n \times n$ chessboard that no two queens attack each other

Explaination :-

1] we keep global array $x[\ldots]$: final answer in n-tuple
index represents k^{th} queen and value at that index represents column on chessboard. Note that k^{th} queen to be placed on k^{th} row, ie. each new queen in new row



2] for each row, we travel for each column, row col
3] we check if that cell is available by place(k,i)
4] if available, note that place for k^{th} queen by $x[k]=i$

5] also if we reach to final queen, print x else travel for ~~next~~ position of next queen

For place(k,i)

1] from initial condition that each queen to be placed on new row, we make sure, no horizontal check required

2] for vertical check, place(k,i)

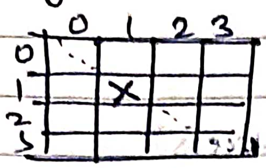
included

for j from 0 to k-1

as this cell not to be taken

$x[j] == i$ → if true then return false

3) diagonal check



place(k,i)

for j in (0, k-1) ← for each previous queen

left diagonal → row − col^n stays same

for previous queens → $\underset{\text{row}}{j} - \underset{\text{col}^n}{x[j]}$

for our current queen which is at (k,i) → k − i

equating both $j - x[j] = k - i$

$j - k = x[j] - i$ —(1) for left diagonal

Right diagonal → row + col^n stays same

previous queens → $j + x[j]$

current queen at (k,i) → $k + i$

equating both $j + x[j] = k + i$

$j - k = i - x[j]$ —(2)

from ~~one~~ (1) & (2)

abs(j−k) = abs(i − x[j]) —(3)

if this comes true then that space is not available

hence x[j] == i & (3) either of them must not be true for place to be available