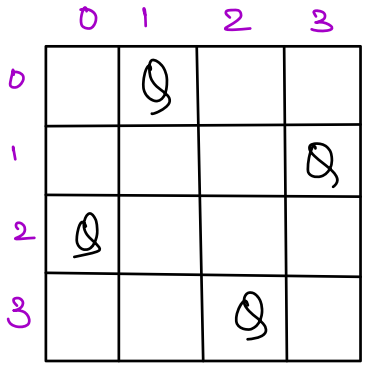
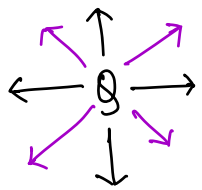
Q.1 **N Queens.**

Amazon
MS:

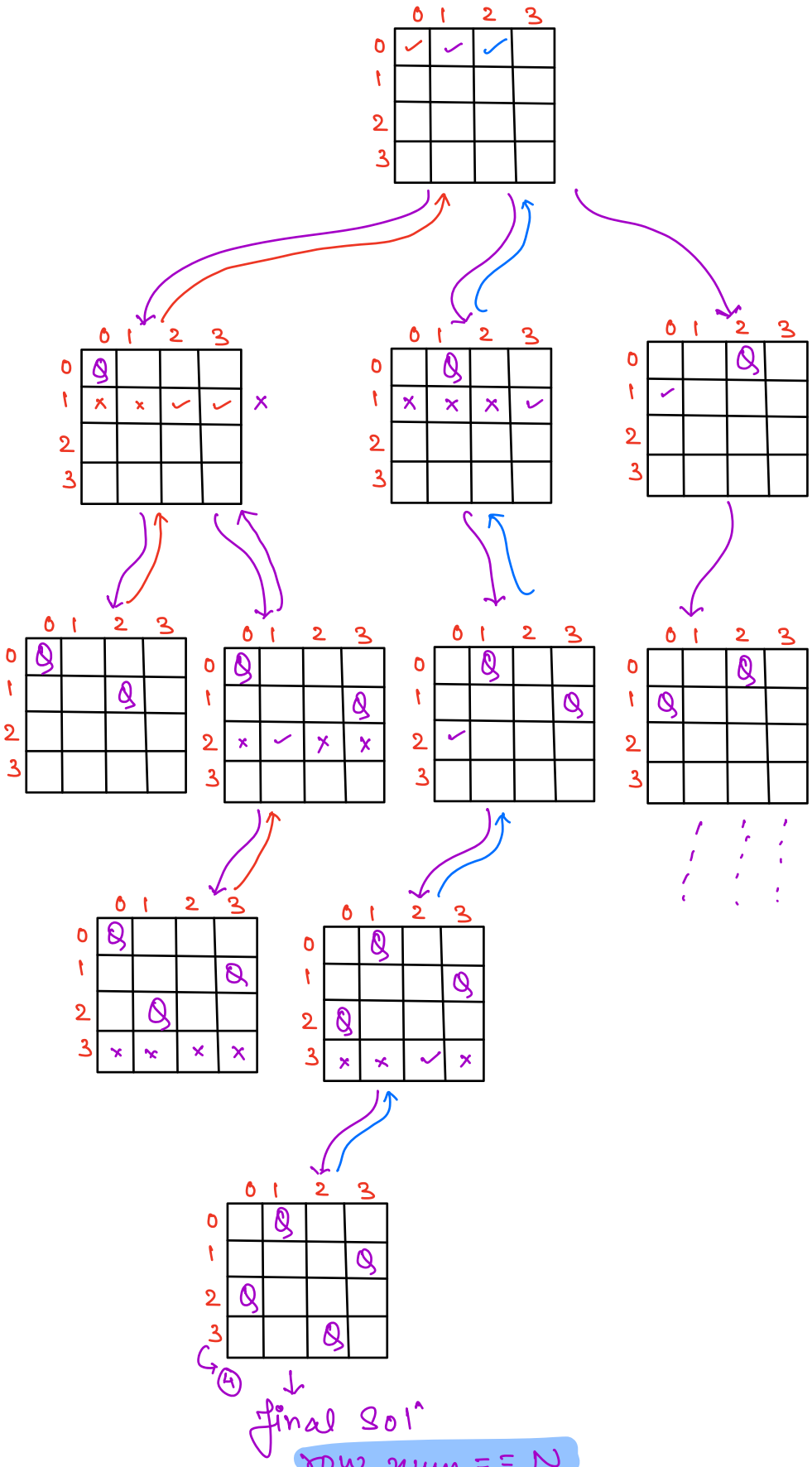
Given N × N matrix, place N queens in such a way that no queen can kill each other.



→ One such possible solution.



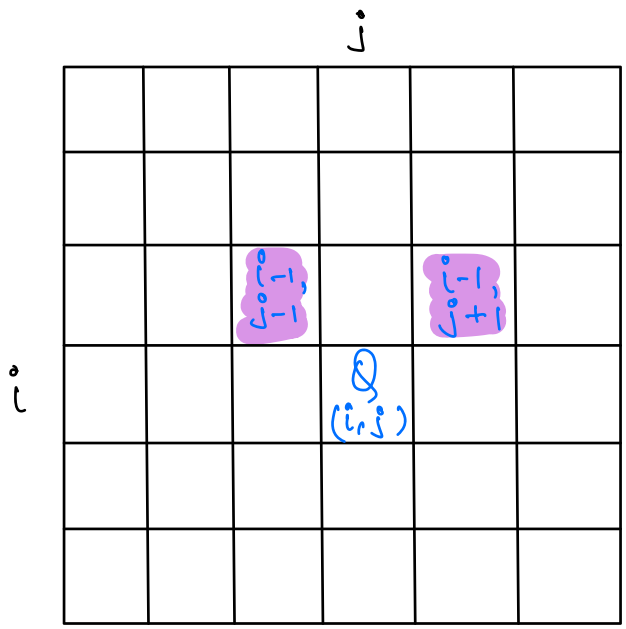
→ Print all possible solutions of N-Queen.


Final Soln
row_num == N

```
void nQueens(int mat[][], N, i){
if(i==N){
        //Print the soln
        return;
}
// At ith row, we need to place a Queen
for(j=0; j<N; j++){
        //Place the Queens at i,j & check
        // if it is possible.
        if(check(mat, i, j)){
                mat[i][j] = 1; //Place Queen
                nQueens(mat, N, i+1);
                mat[i][j] = 0; // Backtrack;
        }
}
```

O(N)

J



```
bool check(mat[][], i, j){
// Check column.
for(r = 0; r < i; r++){
    if(mat[r][j] == 1)
        return false;
}
// Check row.
for(c = 0; c < j; c++){
    if(mat[i][c] == 1)
        return false;
}
r = i-1, c = j+1; // Right Diagonal.
while(r >= 0 && c < N){
    if(mat[r][c] == 1)
        return false;
    r--, c++;
}
r = i-1, c = j-1; // Left Diagonal.
while(r >= 0 && c >= 0){
    if(mat[r][c] == 1)
        return false;
    r--, c--;
}
return true;
}
```

Check (1)

TC: $O(N! \times N)$ (W.C)

HW: Explore more about TC.

SC: $O(N \times N)$

Q.2 SUDOKU.

↳ 9×9 matrix, fill this matrix.

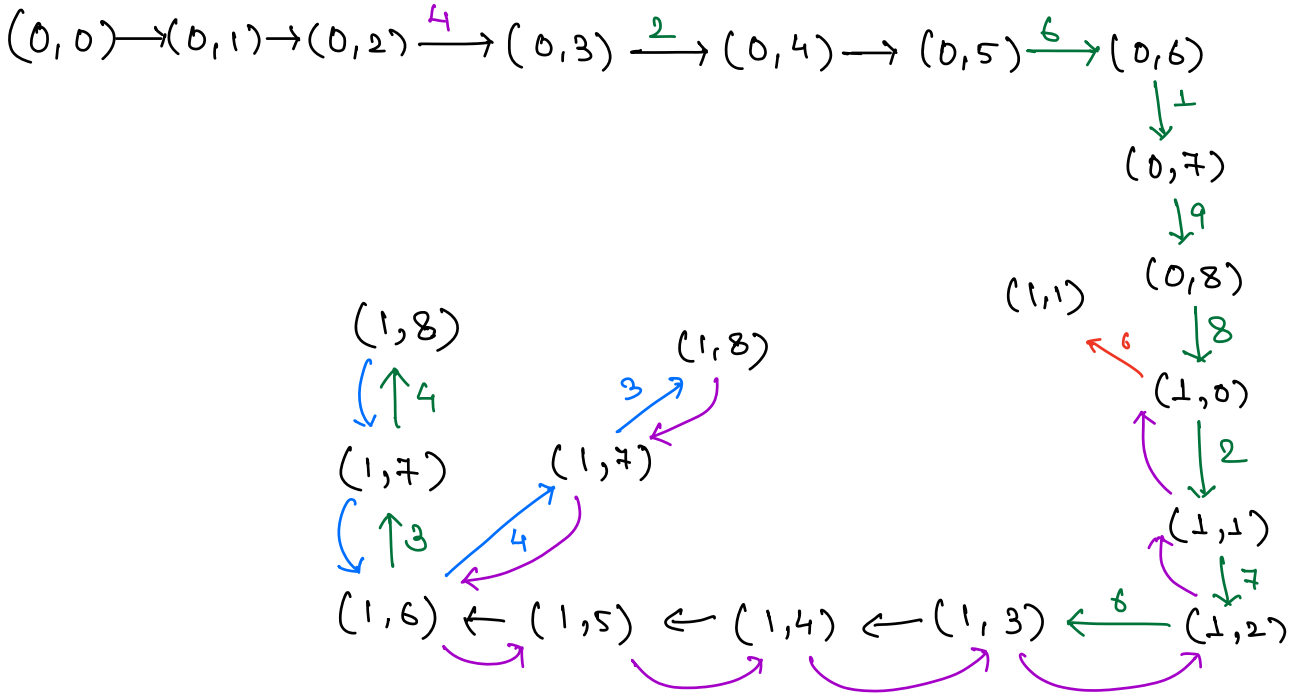
→ Numbers from 1 to 9

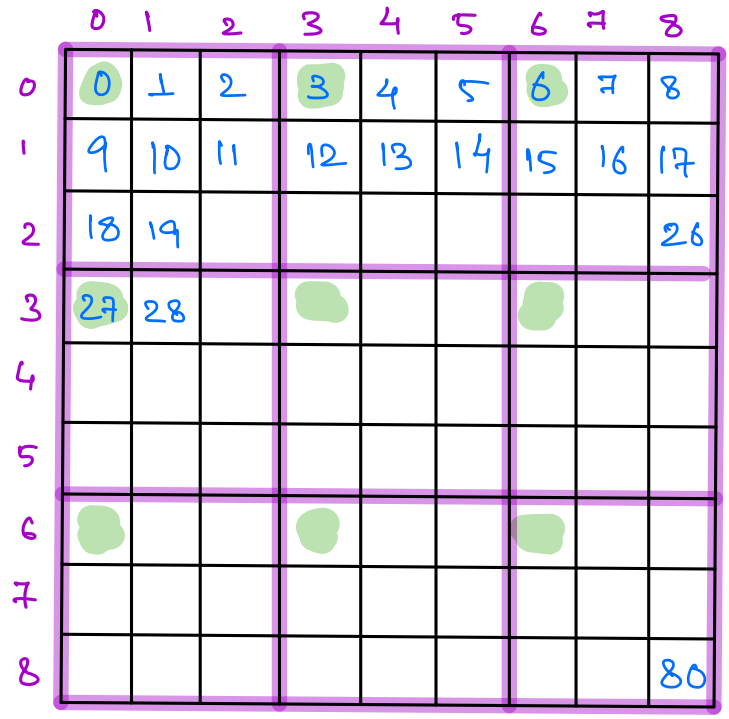
i) → In a row, number can't repeat

ii) → In a col, number can't repeat

iii) → In 3×3 box, number can't repeat

	0	1	2	3	4	5	6	7	8
0	5	3	4	2	7	6	1	9	8
1	6			1	9	5			
2		9	8					6	
3	8		2		6				3
4	4			8		3			1
5	7		3		2				6
6		6					2	8	
7				4		9			5
8			1		8			7	9





9×9

3×3

i, j

14 → 14/9 ⇒ 1 ⇒ row
14 % 9 ⇒ 5 ⇒ col
⇒ (1, 5)

28 → 28/9 = 3
28 % 9 = 1
⇒ (3, 1)

80 ⇒ 80/9 = 9 = 8
80 % 9 = 0 ⇒ 8
⇒ (8, 8)

start of Box

1, 4 → 0, 3

2, 7 → 0, 6

4, 4 → 3, 3

7, 7 → 6, 6

7, 2 → 6, 0

5, 2 → 3, 0

$x \longrightarrow x - x\%3$

```
void solveSudoku(mat[][], n)
    if (n == 81) {
        // Print the matrix
        return;
    }
    r = n/9 , c = n%9
    if(mat[r][c] != 0){
        solveSudoku(mat, n+1);
    }
    else{
        for(i = 1; i<=9; i++){
            if(valid(mat,r,c,i)){
                mat[r][c] = i;
                solveSudoku(mat, n+1);
                mat[r][c] = 0;
            }
        }
    }
}
```

```
bool valid (mat[], r, c, d) {
    for (i = 0; i < 9; i++) {
        if (mat[r][i] == d)        → Check r^th row
            return false;
        if (mat[i][c] == d)        → Check c^th col
            return false;
    }

    // 3x3 Box
    x = r - r%3
    y = c - c%3
    for (i = x; i < x+3; i++) {
        for (j = y; j < y+3; j++) {
            if (mat[i][j] == d)
                return false;
        }
    }
    return true;
}
```

TC: $(9 \times 9 \times 9 \times \dots 9) \Rightarrow 9^{81}$

81 times.

$\Rightarrow 9^{n^2}$ [n = 9]

$\Rightarrow n^{n^2}$

SC: $O(81) \simeq O(1)$
