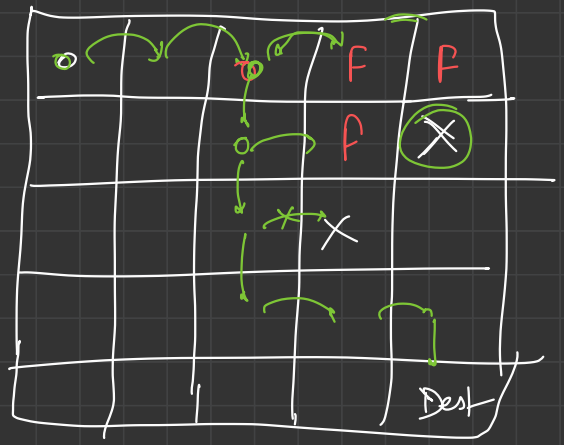
Rat in a maze : rightmost path

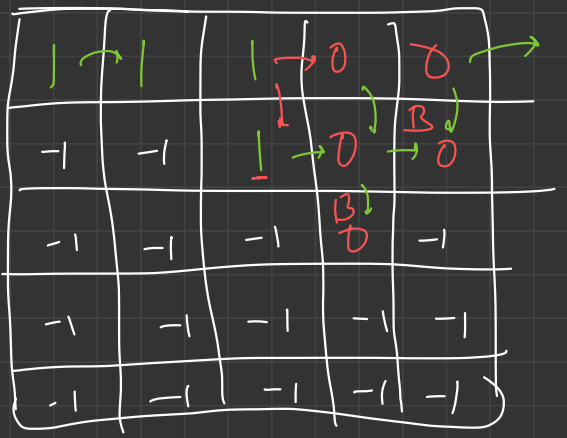
① using visited to avoid repeated func. call for same cell.

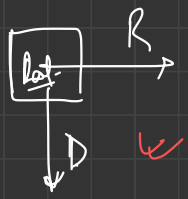
② printing all possible paths.

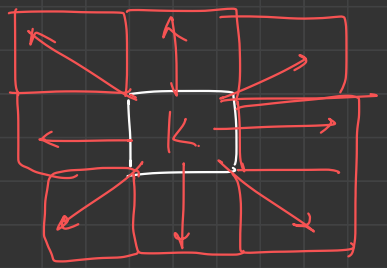
0 → no path exist
1 → path to dest exist
-1 → unvisited



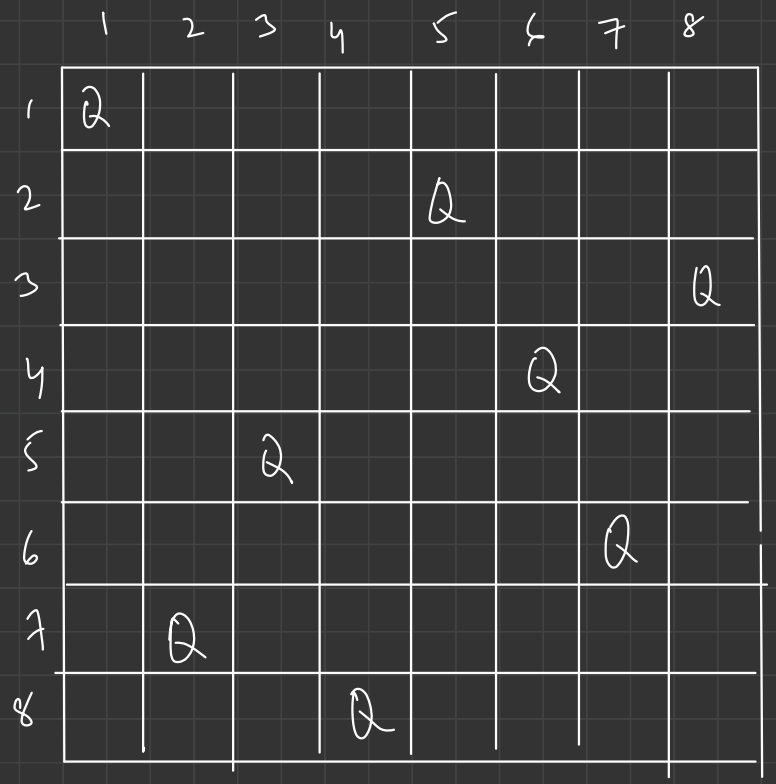
visited







8 possible steps explore

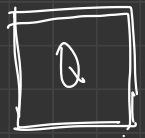


row ✓
col ✓

no queens attack each other

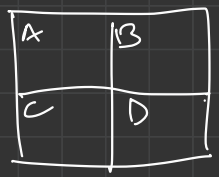
n=1

n x n → n queens place



→ 1 sol

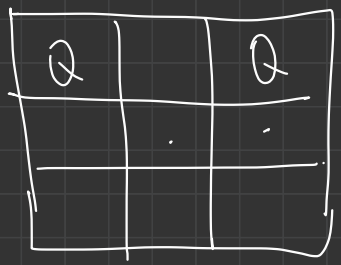
n=2



??

→ no soln

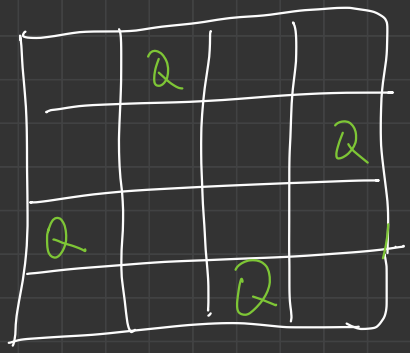
n=3

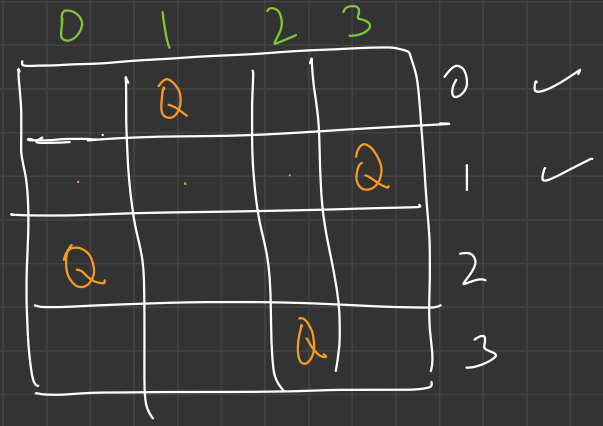


3-queens place

→ no Soln

$n = 4$



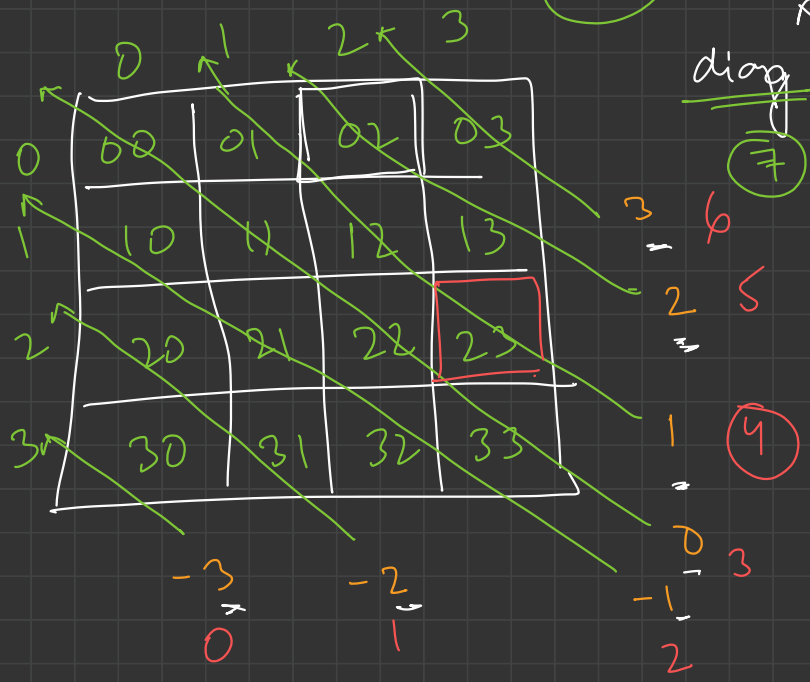


row → 0 cols ~~0~~, ~~1~~, 2, 3

row → 1 cols ~~0~~ ~~1~~ ~~2~~ 3

row → 2 cols 0 ~~1~~ ~~2~~ ~~3~~

row → 3 col



row → 4 → all rows filled successfully

↳ a solution ⇒ count++

n = 4

diag1

(7)

col - row + 3

cell (2 3)

col - row

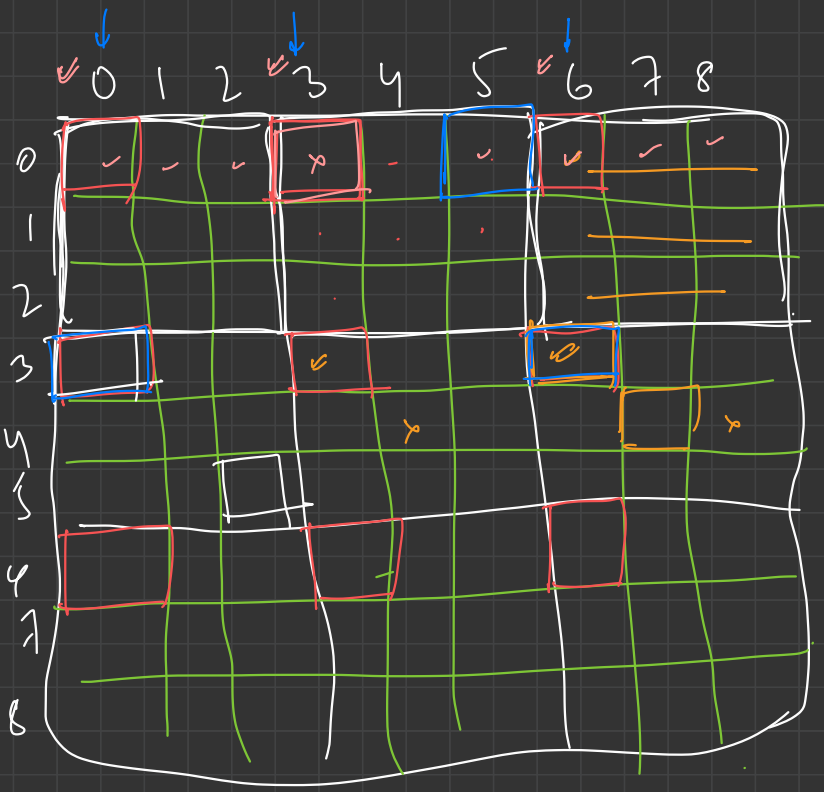
3 - 2 = 1

(4)

cell (0 2)

col - row

2 - 0 = (2)



(i, j) → digit

① find the sub-grid of (i, j)

↳ top-left cell of that sub-grid

S % 3

↳ dist from last multiple of 3

② once, you have (subRow, subCol)

Check 3x3 matrix starting from that index

7 % 3

↳ dist of 7 from last multiple of 3

(i, j) : subgrid_startRow = i - (i % 3)

subgrid_startCol = j - (j % 3)

(4,7): startRow = 4 - 4%3

= 4-1 = 3

startCol = 7 - 7%3

= 7-1 = 6

(3,6) → subgrid start cell

(5,2): startRow = 5 - 5%3

= 5-2 = 3

startCol = 2 - 2%3

= 2-2 = 0

(3,0) → start of subgrid

pair

5	2	6	1
2	1	1	0

$\Rightarrow$ (4) → output

find no. of pairs $(a[i], a[j])$ such that

$i < j$ and

$a[i] > a[j]$

- (5, 2)
- (5, 1)
- (2, 1)
- (6, 1)

(4)

⇒ 1 2 5 6 | 1 1 2 3

sorted sorted

sorted array ⇒ no. of inversions ?! = 0

2 → 1 1

5 → 1 1 2 3

6 → 1 1 2 3

10 pairs

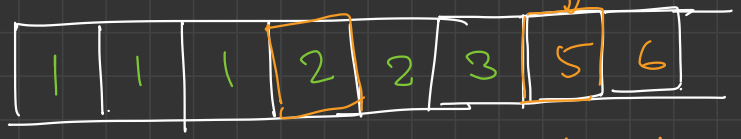
merge

left: * 2 5 6 → 0 1 2 3

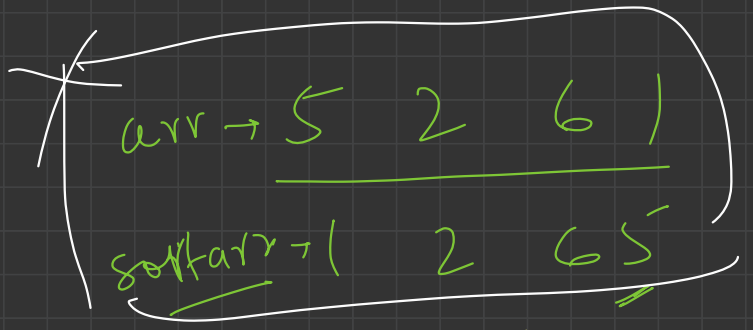
right: * * 2 3 → 0 1 2 3

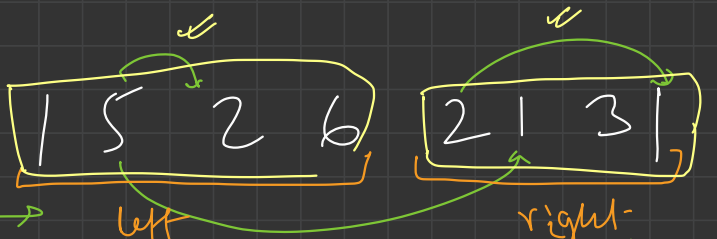
i = 2

j = 4



j → 2 4 4 ⇒ 10





prob- no. of inverted pairs

have to count all such inverted pairs:

$a[i] > a[j]$

for $i < j$

① no. of IP in left {5,2}

② no of IP in right {2,1}

③ no of IP such that first element is in left & {5,1}

second element is in right

① and ② are sub-problems : solve recursively

③ can be found using merge by sorting ① and ②

total answer = ① + ② + ③