B.1) Given a row-wise and column-wise sorted matrix, find if the element k is present or not.

$$A = \begin{bmatrix} -5 & -2 & 1 & 13 \\ -4 & 0 & 3 & 14 \\ -3 & 2 & 6 & 18 \end{bmatrix}$$

$k = 13 \Rightarrow$ ✓

$k = 15 \Rightarrow$ ✗

$k = 2 \Rightarrow$ ✓

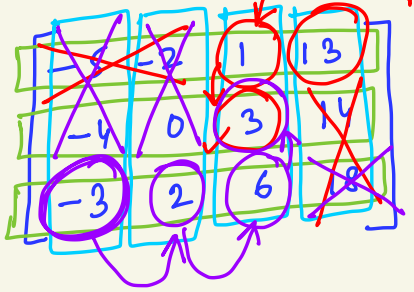
$k = 3$

dec

inc

$3 < 13$

$3 > 1$



dec

inc.

$3 > -3$

$3 > 2$

$3 < 6$

$$\begin{bmatrix} 1 & 2 & 5 \\ 2 & 3 & 7 \\ 3 & 4 & 9 \end{bmatrix}$$

Start at

Top right ⇒ ↓

← ⇒ $(m-1)$ times

↓ ⇒ $(n-1)$ times

$(n+m-2)$ times [Max no. of comparisons / steps]

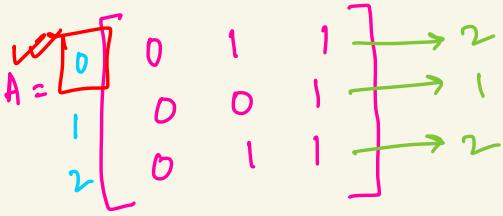
```
i = 0 , j = m-1
while (i < n && j >= 0) {
    if (A[i][j] == k)
        return true
    if (A[i][j] < k)
        i++
    else
        j--
}
return false
```

O(n+m) T.C.
O(1) S.C

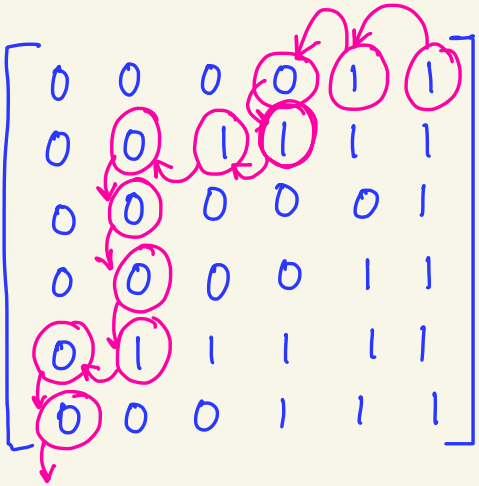
B.2) Given a binary sorted matrix A of size n * n. Find the row with most 1s.

↓ each row is sorted

⇓ If multiple rows, then return the lower index row.



Ans → 0.



0 → ~~1~~ 2

1 → ~~3~~ 4

4 → 5

O(n) T.C.

Max n steps down, max n steps left.

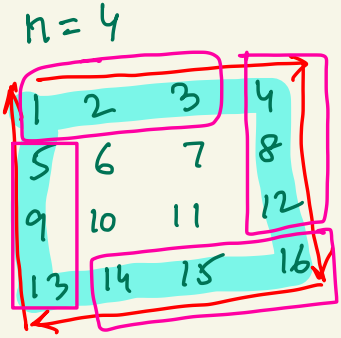
n + n = (2n) steps

```
i = 0, j = n-1, ans = 0
while (i < n && j >= 0) {
    if (A[i][j] == 0)
        i++
    else {
        ans = i
        j--
    }
}
return ans
```

O(n) T.C.

O(1) S.C.

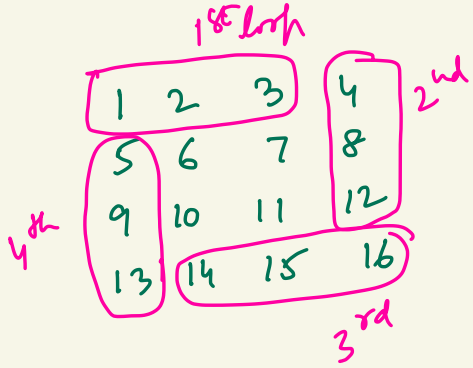
Q 3) Print Boundary Elements of a matrix n*n



O/P → 1 2 3 4 8 12 16 15 14 13 9 5

→ Print (n-1) elements of 1st row L → R
→ Print (n-1) elements of last col T → B
→ Print (n-1) elements of last row R → L
→ Print (n-1) elements of 1st col B → T

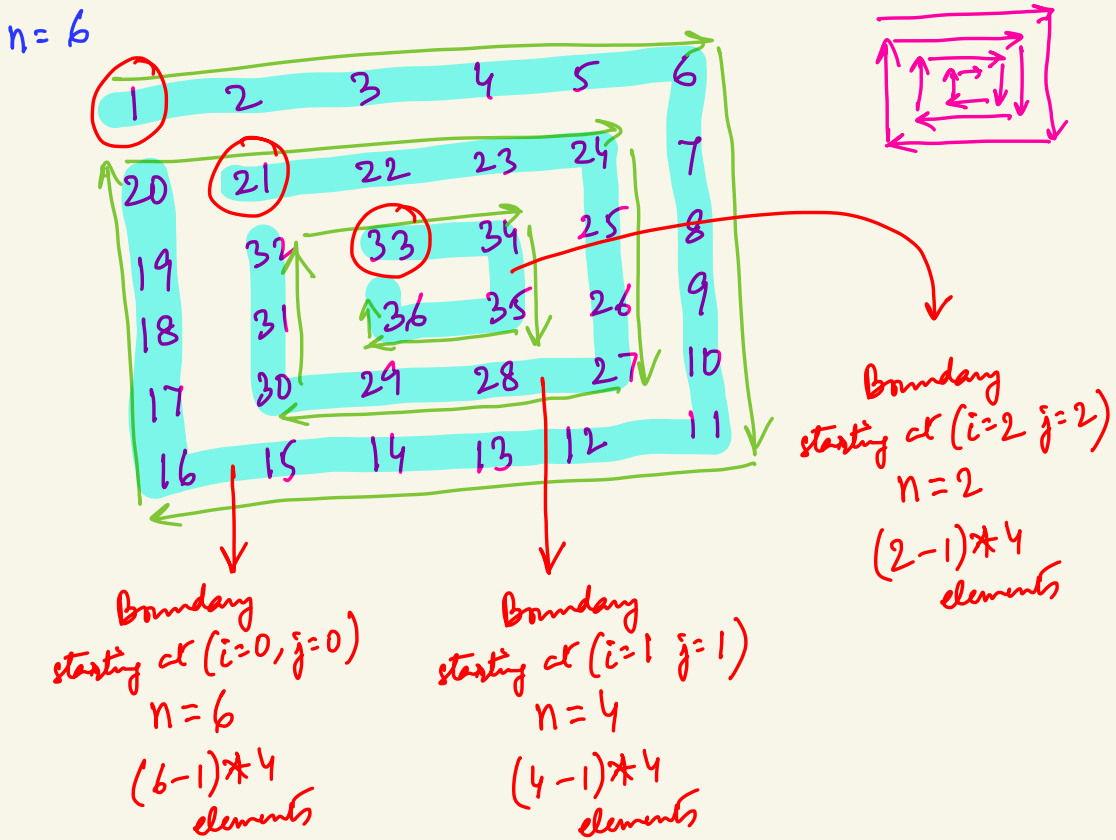
```
i=0, j=0
// Top row (n-1) L→R
for(idx=0; idx<n-1; idx++){
    Print(A[i][j])
    j++
}
// Last col (n-1) T→B
for(idx=0; idx<n-1; idx++){
    Print(A[i][j])
    i++
}
// Btm row(n-1) R→L
for(idx=0; idx<n-1; idx++){
    Print(A[i][j])
    j--
}
// 1st col (n-1) B→T
for(idx=0; idx<n-1; idx++){
    Print(A[i][j])
    i--
}
```



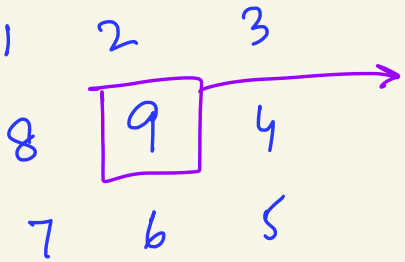
O(n) T.C

O(1) S.C

Q4) Given an integer A, generate a spiral matrix with elements 1 to A^2 in spiral order and return the generated matrix.



n = 3

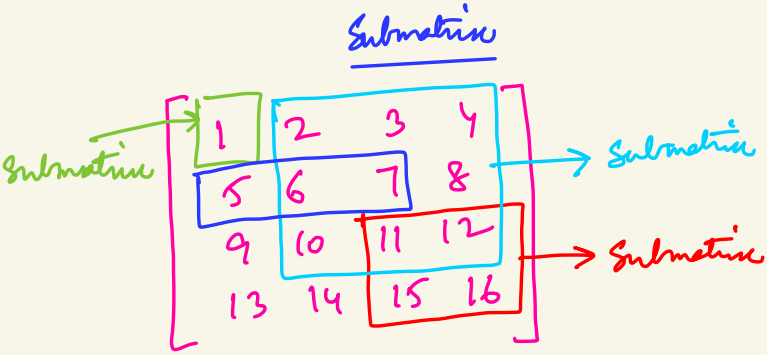


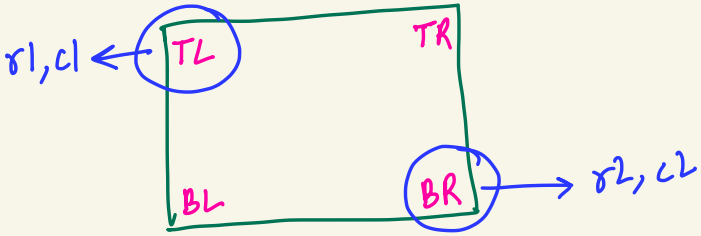
```
int ans[A][A]
val = 1
i = 0, j = 0
while (A > 1) {
    for (idx = 1; idx < A; idx++) {
        ans[i][j] = val
        j++
        val++
    }
    for (idx = 1; idx < A; idx++) {
        ans[i][j] = val
        i++
        val++
    }
    for (idx = 1; idx < A; idx++) {
        ans[i][j] = val
        j--
        val++
    }
    for (idx = 1; idx < A; idx++) {
        ans[i][j] = val
        i--
        val++
    }
    i++
    j++
    A -= 2
}
if(A==1)
    ans[i][j] = val
return ans
```

$O(n^2)$ T.C

$O(1)$ extra space.

[Break till 10:48 PM]


Submatrix
Submatrix
Submatrix
Submatrix
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16


r1, c1
TL
TR
BL
BR
r2, c2

BS) Given a matrix A[n][m], determine the sum of all possible submatrices.

$$\begin{array}{c} \\ 0 \\ 1 \end{array}\begin{array}{c} \begin{array}{ccc} 0 & 1 & 2 \end{array} \\ \begin{bmatrix} 4 & 9 & 6 \\ 5 & -1 & 2 \end{bmatrix} \end{array}$$

$[4] \to 4$
$[9] \to 9$
$[6] \to 6$
$[5] \to 5$
$[-1] \to -1$
$[2] \to 2$

$[4\ 9] \to 13$
$[9\ 6] \to 15$
$[5\ -1] \to 4$
$[-1\ 2] \to 1$
$[4\ 9\ 6] \to 19$
$[5\ -1\ 2] \to 6$

$\begin{bmatrix} 4 & 9 \\ 5 & -1 \end{bmatrix} \to 17$
$\begin{bmatrix} 9 & 6 \\ -1 & 2 \end{bmatrix} \to 16$
$\begin{bmatrix} 4 & 9 & 6 \\ 5 & -1 & 2 \end{bmatrix} \to 25$

$\begin{bmatrix} 4 \\ 5 \end{bmatrix} \to 9$
$\begin{bmatrix} 9 \\ -1 \end{bmatrix} \to 8$
$\begin{bmatrix} 6 \\ 2 \end{bmatrix} \to 8$

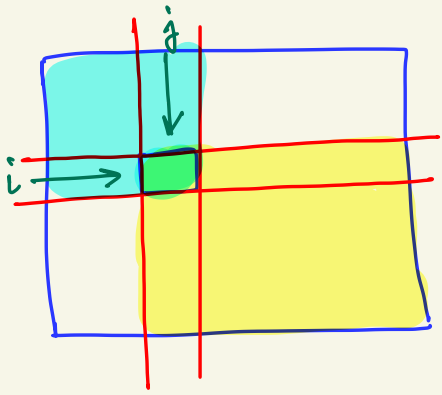
Total sum → 166.

No. of submatrices in A[n][m] → $\left[\frac{n(n+1)}{2} * \frac{m(m+1)}{2}\right]$

sum → $4*\underline{6} + 9*\underline{8} + 6*\underline{6} + 5*\underline{6} + (-1)*\underline{8} + 2*\underline{6}$

$= \sum \left(A[i][j] * \text{No. of submatrices having } (i,j) \right)$

Contribution Technique



$r_1 \in [0, i] \Rightarrow i+1$

$r_2 \in [i, n-1] \Rightarrow n-i$

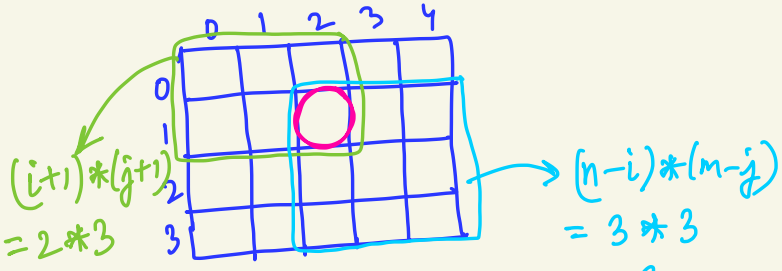
$c_1 \in [0, j] \Rightarrow j+1$

$c_2 \in [j, m-1] \Rightarrow m-j$

(r_1, c_1) TL $\quad$ (r_2, c_2) BR

submatrices having (i,j) $\longrightarrow$ $(i+1)*(n-i)*(j+1)*(m-j)$

$$\text{Total sum} = \sum_{i \in [0,n-1]} \sum_{j \in [0,m-1]} \left(A[i][j] * (i+1)*(n-i)*(j+1)*(m-j) \right)$$



$6 * 9 = 54.$

```
total = 0
for (i → 0 to n-1) {
    for (j → 0 to m-1) {
        total += A[i][j] * (i+1) * (n-i) * (j+1) * (m-j)
    }
}
return total
```

$O(n*m)$ T.C.

$O(1)$ S.C.