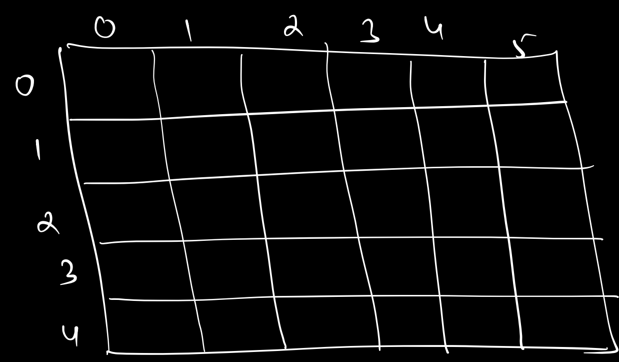
NOTE :- Max Subarray Sum → TLE → (why) → next class

int mat[5][6]

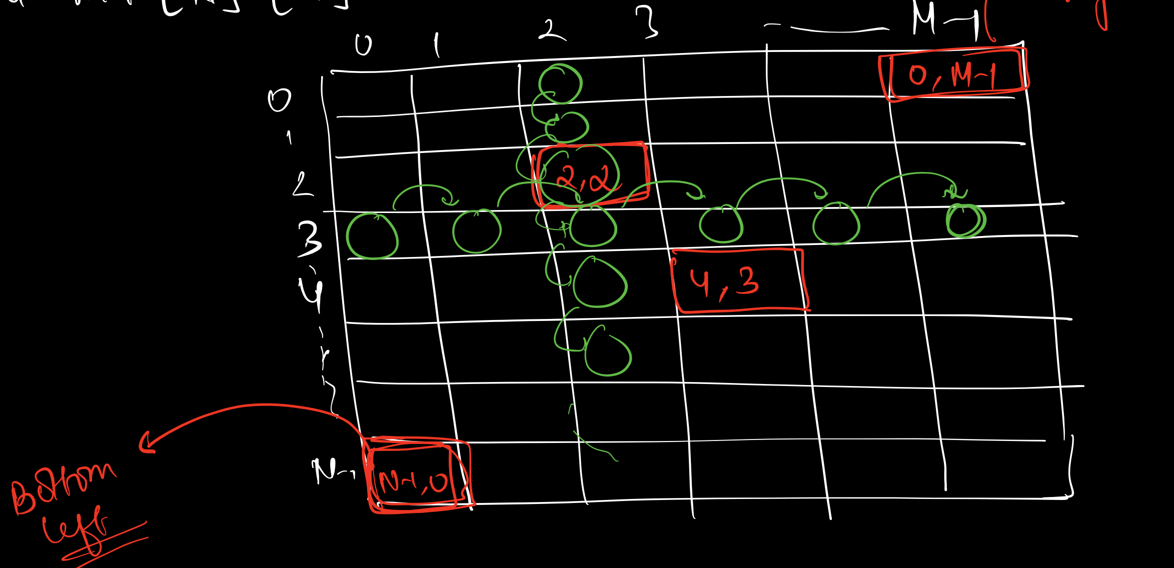
↓ rows ↓ columns



rows ↑ col ↑

int mat[N][M]

top right



mat[2][2]   mat[4][3]

mat[N-1][0]   mat[0][M-1]

i → row

j → colm

→ iterate on a row :- [3]

```
for(j=0; j<M; j++){
    print( mat[3][j] )
}
```

→ iterate on a column : [2] :

```
for( i=0; i<N; i++){
        print( mat [i] [2])
}
```

Q.1. Given a mat[N][M], print row-wise sum.

ex ⇒ mat[3][4] ⇒

$$\begin{array}{c|cccc|l} & 0 & 1 & 2 & 3 & \\ \hline 0 & 3 & 8 & 9 & 2 & \to 22 \\ 1 & 1 & 2 & 3 & 6 & \to 12 \\ 2 & 4 & 10 & 11 & 17 & \to 42 \\ \hline \end{array}$$

O/P ⇒ 22, 12, 42.

pseudo

```
for( i=0; i<N; i++){
    sum = 0;
    for( j=0; j<M; j++){
        sum = sum + mat [i] [j]
    }
    print(sum)
}
```

$$\boxed{TC \Rightarrow O(N \times M) \; ; \quad SC \Rightarrow O(1).}$$

Q2. Given a mat[N][M], find max column sum.

mat[3][4] =

3	8	9	2
1	2	3	6
4	10	11	8
↓	↓	↓	↓
8	20	23	16

max col sum = 23   o/p = 23

pseudo

```
maxAns = INT_MIN;
for( j=0; j<M; j++){
    sum = 0
    for(i=0; i<N; i++){
        sum = sum + mat[i][j]
    }
    maxAns = max(maxAns, sum)
}
print(maxAns).
```

$\boxed{\begin{array}{l} TC \Rightarrow O(NM) \\ SC \Rightarrow O(1) \end{array}}$

Q3. Given a mat[N][N], print its diagonals. → L to R
→ R to L

ex ⇒ $\underbrace{\text{mat[4][4]}}_{\text{sq. matrix}}$ ⇒

	0	1	2	3
0	0,0			0,3
1		1,1	1,2	
2		2,1	2,2	
3	3,0			3,3

⇒ // print L-R

0,0 → 1,1 → 2,2 → 3,3.

i, j

i++   i++   i++

j++   j++   j++

pseudo

```
i = 0
while( i<N ){
    print( mat[i][i] )
    i++
}
```

TC ⇒ O(N)

SC ⇒ O(1)

// print R-L

0,3 → 1,2 → 2,1 → 3,0.

i j   i j   i j   i j

i++   i++   i++

j--   j--   j--

```
i=0; j = N-1
while( i<N && j>=0 ){
    print( mat[i][j] )
    i++
    j--
}
```

TC ⇒ O(N)

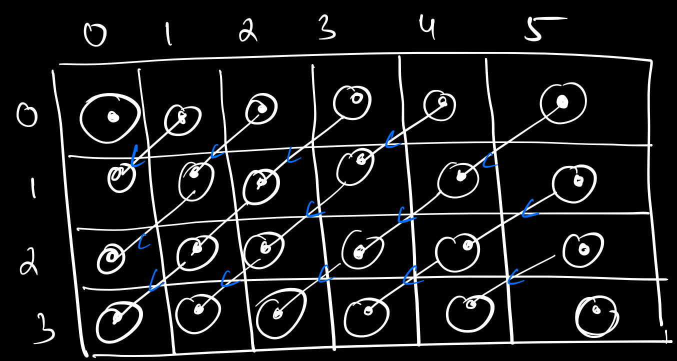
SC ⇒ O(1)

overall ⇒

$$\boxed{\begin{array}{l} TC: O(2N) \simeq O(N) \\ SC: O(1) \end{array}}$$

Q4. Given mat[N][M], print all diagonals [R-L]

mat[4][6] ⇒



Obs → all diagonals start from 0th row or, M-1th colm

ex ⇒

	0	1	2	3	4
0	1	2	3	4	5
1	6	7	8	9	10
2	11	12	13	14	15

O/P ⇒

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

lets start from cell

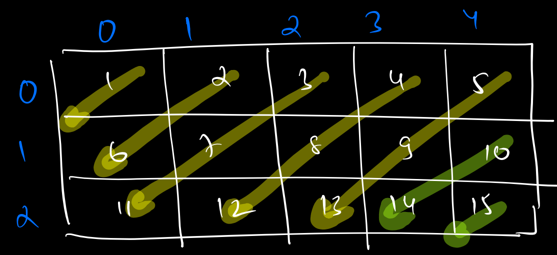
i j

$\langle 0,3\rangle$ $\langle 0,2\rangle$ $\langle 0,1\rangle$

i++ $\langle 1,2\rangle$ j-- $\langle 1,1\rangle$ i++ $\langle 1,0\rangle$ j--

i++ $\langle 2,1\rangle$ j-- $\langle 2,0\rangle$ ~~$\langle 2,-1\rangle$~~

~~$\langle 3,0\rangle$~~   ~~$\langle 3,-1\rangle$~~



* important to keep both row & colm inside bound.

* two conditions :- start → $i = 0$ && $j = M-2$ ← start
$i < N$ $j \geq 0$

pseudo

S1 → print all diagonals for 0^{th} row.

S2 → print all diagonals for $M-1^{th}$ col.

S1 ⇒ all diagonals for 0^{th} row :-

```
for(j=0; j<M-1; j++){
    x=0      ← row
    y=j      ← colm
    while( x<N && y>=0){
        print(mat[x][y])
        x++, y--
    }
}
```

op

1
2 6
3 7 11
4 8 12

j = ~~0~~ ~~1~~ ~~2~~ 3

x	y

	0	1	2	3	4
0	1	2	3	4	5
1	6	7	8	9	10
2	11	12	13	14	15

```
for ( i = 0; i < N; i++ ) {
    x = i
    y = M-1
    while ( x < N && y >= 0 ) {
        print( mat[x][y] )
        x++
        y--
    }
}
```

OP

5 9 13

10 14

15

i = ~~0~~ ~~1~~ 2

x	y
2	4
~~3~~	~~3~~

	0	1	2	3	4
0	1	2	3	4	5
1	6	7	8	9	10
2	11	12	13	14	15

$$\boxed{\begin{aligned} TC &\Rightarrow O(N \times M) \\ SC &\Rightarrow O(1) \end{aligned}}$$

Q5:, Given a mat[N][N], find the transpose inplace.

i) transpose ⇒ flipping rows ⇄ colm

1	2	3
4	5	6
7	8	9

$\xrightarrow{\text{transpose}}$

1	4	7
2	5	8
3	6	9

ii) inplace ⇒ no extra memory
update the i/p matrix.

// mat[3][4]

1	2	3	4
5	6	7	8
9	10	11	12

3x4

→

1	5	9
2	6	10
3	7	11
4	8	12

4x3

So, important ⇒ if given i/p is not a sq. matrix,
inplace transpose is not possible

eg ⇒ mat[5][5] ⇒

	0	1	2	3	4
0	1	2	3	4	5
1	6	7	8	9	10
2	11	12	13	14	15
3	16	17	18	19	20
4	21	22	23	24	25

transpose

$$
\begin{array}{c|ccccc}
 & 0 & 1 & 2 & 3 & 4 \\
\hline
0 & 1 & 6 & 11 & 16 & 21 \\
1 & 2 & 7 & 12 & 17 & 22 \\
2 & 3 & 8 & 13 & 18 & 23 \\
3 & 4 & 9 & 14 & 19 & 24 \\
4 & 5 & 10 & 15 & 20 & 25
\end{array}
$$

Obs

1) diag. (L-R) is same in both

2) swapping elements across the diagonal :-

swap → mat[0][1] ⇒ mat[1][0]

swap → mat[3][2] ⇒ mat[2][3]

mat[3][4] ⇒ mat[4][3].

generalise ⇒ mat[i][j] ⇒ mat[j][i].

pseudo

```
for(i=0; i<N; i++){
    for(j=0; j<N; j++){
        swap(mat[i][j], mat[j][i])
    }
}
```

wrong → brings original matrix back

when → reach → [0,3]
↓
swap ([0,3], [3,0])

reach → [3,0]
↓
swap ([3,0], [0,3]).

So, either iterate on upper triangle & swap.
or, " " lower "

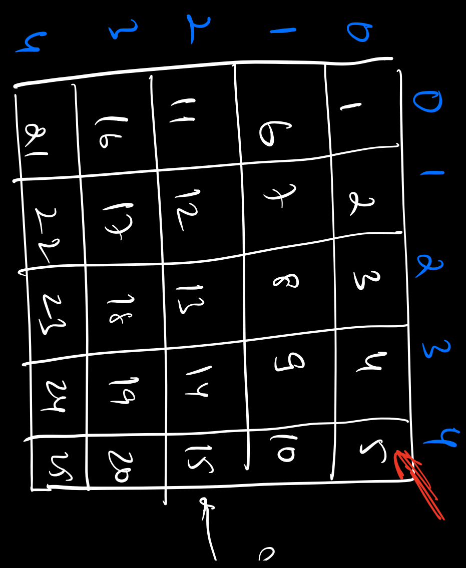
pseudo → How [next class]

TC ⇒ O(?) → O($N^2/2$) → O(N^2)
SC ⇒ O(1)

Q6. Given a sq. matrix, rotate it by 90° in clockwise
w.r.t TR in place.

	0	1	2	3	4
0	1	2	3	4	5
1	6	7	8	9	10
2	11	12	13	14	15
3	16	17	18	19	20
4	21	22	23	24	25

rotation →



↓ transpose

	0	1	2	3	4
0	1	6	11	16	21
1	2	7	12	17	22
2	3	8	13	18	23
3	4	9	14	19	24
4	5	10	15	20	25

→ reverse row

↓ same

	0	1	2	3	4
0	21	16	11	6	1
1	22	17	12	7	2
2	23	18	13	8	3
3	24	19	14	9	4
4	25	20	15	10	5

i) Step1 ⇒ find transpose. → TC ⇒ $O(N^2)$

ii) Step 2 ⇒ reverse each row. → TC ⇒ $O(N^2)$

$$\boxed{\begin{array}{l} TC \Rightarrow O(2N^2) \Rightarrow O(N^2) \\ SC \Rightarrow O(1) \end{array}}$$

1) no class on friday → 3/6.

2) Monday → 6/6

↓

Interview probs. on arrays

1	2	3
4	5	6
7	8	9

$\Rightarrow$

1

2 3

4 5 6

7 8

9

1 2 3 4

1 2 3

1　　2　　3

1 2　　2 3

1 2 3

$$\frac{1 + (1+2) + (1+2+3) + (2) + (2+3) + \underline{3}}{(1*3) + (2*4) + (3*3).}$$

1 2 3

4 5 6

7 8 9

$\Rightarrow$

4 1 2

7 5 3

8 9 6