

2D Matrix

→ 2D array

→ Rectangular of numbers, where each number is called an element.

$\text{int mat}[N][M]$
 ↓
 Columns (Vertical)
 ↓
 Rows (Horizontal)

	0	1	2	3	4
0	0,0	0,1	0,2	0,3	0,4
1	1,0	1,1	1,2	1,3	1,4
2	2,0	2,1	2,2	2,3	2,4
3	3,0	3,1	3,2	3,3	3,4
4	4,0	4,1	4,2	4,3	4,4

$\Rightarrow \text{mat}[5][5]$

$\text{mat}[N][M]$

	0	1	2	...	j	...	M-2	M-1
0	0,0				0,j			0,M-1
1					1,j			
2					2,j			
...					...			
i	i,0	i,1	i,2	...	i,j	...	i,M-2	i,M-1
...					...			
N-2					N-2,j			
N-1	N-1,0				N-1,j			N-1,M-1

Obs: When iterating in a row

→ Row number is fixed

→ Column number goes from 0 to $m-1$

When iterating in a column

→ Column number is fixed

→ Row number goes from 0 to $n-1$

Q Given 2D Matrix, $\text{mat}[n][m]$.

Print the row-wise Sum.

	0	1	2	3	
$\text{mat}[3][4] = 0$	1	2	3	4	$\Rightarrow 10$
1	5	6	7	8	$\Rightarrow 26$
2	9	10	11	12	$\Rightarrow 42$

```
void sumRow (int mat[n][m]) {
```

```
    int N = mat.size()
```

```
    int m = mat[0].size()
```

TC: $O(n \times m)$

SC: $O(1)$

```
    for (i = 0; i < N; i++) {
```

```
        int sum = 0;
```

```
        for (int j = 0; j < m; j++) {
```

```
            sum += mat[i][j]
```

```
        }
```

```
        print (sum)
```

```
    }
```

```
}
```

Q Given 2D Matrix, $mat[N][M]$.
Print the row-wise sum.

$mat[3][4] =$

	0	1	2	3
0	1	2	3	4
1	5	6	7	8
2	9	10	11	12
	15	18	21	24

```
void colSum(int mat[N][M]){
```

```
    int N = mat.size()
```

```
    int M = mat[0].size()
```

TC: $O(N \times M)$

SC: $O(1)$

```
    for(int j=0; j<M; j++){
```

```
        int sum = 0;
```

```
        for(int i=0; i<N; i++){
```

```
            sum += mat[i][j]
```

```
        }
        print(sum)
```

```
    }
```

```
}
```

Q Given a 2D Square matrix.

Print all the diagonals from left to right.

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

How many diagonals in a square matrix? $\Rightarrow 2$

Principal (Right)
Diagonal

$\Rightarrow 1 \ 5 \ 9$

\Rightarrow Only Printing Principal Diagonal

Row number ==
Col number

```
void printDiagonal(int mat[N][N]) {
```

```
    int i = 0
```

```
    while(i < N) {
```

```
        print(mat[i][i]);
```

```
        i = i + 1;
```

```
    }
```

(0, 0)

↓

(1, 1)

↓

(2, 2)

↓

(3, 3)

↓

(N-1, N-1)

TC: $O(N)$

SC: $O(1)$

⇒ Print Anti-Diagonal

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

Anti-Diagonal (Left)

⇒ 3 5 7

(0, 2)
↓
(1, 1)
↓
(2, 0)

Row += 1
Col -= 1

```
void printAntiDiagonal ( int mat[N][N]) {
```

```
    int i=0, j=N-1;
```

```
    while(i < N) { // we can also use j >= 0
```

```
        print(mat[i][j]);
```

```
        i++;
```

```
        j--;
```

```
    }
```

```
}
```

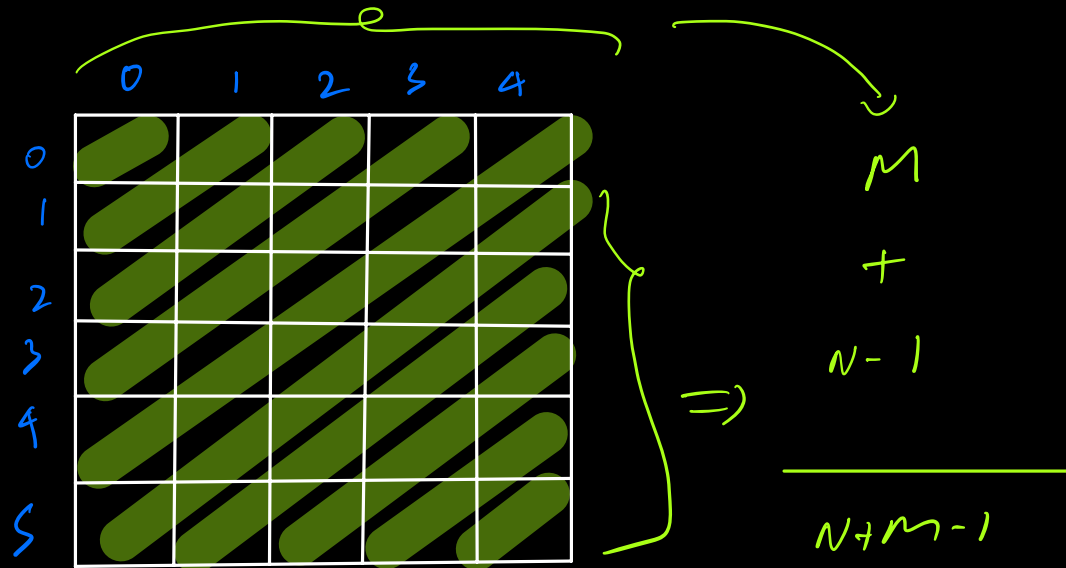
TC: $O(N)$

SC: $O(1)$

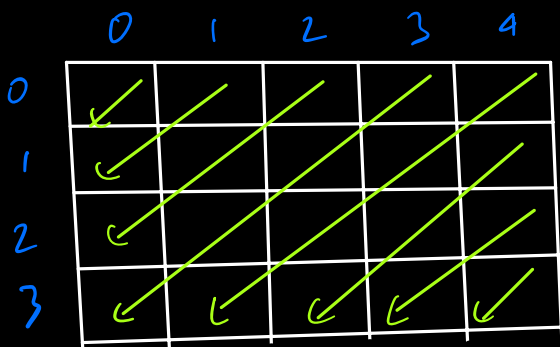
Q Print all anti-diagonals in a non-square matrix?

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

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



Q Given 4×5 matrix. How many right to left diagonals will be there?



$5 \Rightarrow$ first Row
 $4 - 1 = 3 \Rightarrow$ last Column

 8

$(0,0)$

$(0,1) \rightarrow (1,0)$

$(0,2) \rightarrow (1,1) \rightarrow (2,0)$

$(0,3) \rightarrow (1,2) \rightarrow (2,1) \rightarrow (3,0)$

row ++

col --

```
void printAllAntiDiagonal ( int mat[N][M] ) {
```

First Row

```
for ( j = 0 ; j < M ; j++ ) {
```

```
    n = 0
```

```
    c = j
```

```
    while ( n < N && c >= 0 ) {
```

```
        print ( mat[n][c] );
```

```
        n++ ;
```

```
        c-- ;
```

```
    }
```

```
    print ( '\n' )
```

```
}
```

Last Col

1 as first row is done
for (~~i = 0~~ ; i < N ; i++) {

```
    n = i
```

```
    c = M - 1
```

```
    while ( n < N && c >= 0 ) {
```

```
        print ( mat[n][c] );
```

```
        n++ ;
```

```
        c-- ;
```

```
    }
```

```
    print ( '\n' );
```

```
}
```

	0	1	2	3
0	✓			
1		✓		
2			✓	

(0,0) → (1,-1)

(0,1) → (1,0) → (2,-1)

(0,2) → (1,1) → (2,0) → (3,-1)

(0,3) → (1,2) → (2,1) → (3,0)

$(1, 3) \rightarrow (2, 2) \rightarrow (3, 1)$

$(2, 3) \rightarrow (3, 2)$

$T.C: O(N * m)$

$SC: O(1)$

Q Given a square 2D matrix $mat[N][N]$, Find transpose.

!!
only valid
for square
matrix.

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

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

Ans

	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

	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

Brute Force: Create a new matrix
and fill each column referring
row given matrix

$T.C: O(N * N)$

$SC: O(N^2)$

Can you do in SC: $O(1)$, change the given matrix.

Obs: \rightarrow Diagonals are same

	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

	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

\rightarrow Considering diagonal as mirror
rest of things are swapped

$$\text{mat}[i][j] \Rightarrow \text{mat}[j][i]$$

void transpose(matrix[N][M]) {

for(int i=0; i<N; i++) {

for(int j=0; j<M; j++) {

int temp = mat[i][j]

mat[i][j] = mat[j][i]

mat[j][i] = temp

}

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

→ This does not work as swapping all cells will undo the first swaps.

Sol: → Consider only upper triangle or lower triangle

```
void transpose(matrix[N][M]) {  
    for(int i = 0; i < N; i++) {  
        for(int j = i+1; j < M; j++) {  
            int temp = mat[i][j]  
            mat[i][j] = mat[j][i]  
            mat[j][i] = temp  
        }  
    }  
}
```

TC: $O(N^2)$

SC: $O(1)$

Q Given a matrix $[n][n]$. Rotate it to 90 degree clockwise

0 1 2 \swarrow 90°

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

=>

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

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

=>

	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

||
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

Each Row =>
Kon

	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

```

rotate(mat[N][M]){
    mat = transpose(mat);
    for(i=0; i<N; i++){
        reverse(mat[i]);
    }
    return matrix
}

```

Doubts

180° rotation

52	62	52	22	12
02	61	81	61	91
51	61	51	21	11
01	6	8	6	9
5	6	5	2	1

	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

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

↳ Rows swapped

0 \Rightarrow N-1

1 \Rightarrow N-2

↳ Reverse each row

1 2 3
4 5 6
7 8 9

7 4 1
8 5 2
9 6 3

0 1 2 3 4 5 6 7 8 9
1 2 3 4 5 6 7 8 9 10

$K=5$

$0 \Rightarrow 5$

$1 \Rightarrow 5$

$2 \Rightarrow 5$

$3 \Rightarrow 5$

$4 \Rightarrow 5$

$5 \Rightarrow 5$

30

$$10 \times 10 = 100$$

$$\frac{10 \times 10}{4}$$

$$N = 10^3$$

$$\frac{10^3 \times 10^3}{4} = O(N^2)$$

$$\approx 10^5$$

$O(N^2)$ not $O(N)$