

## Today's Content

- 1) 2D Matrix Intro
  - 2) Matrix Traversal Problems
- 

Till Now :

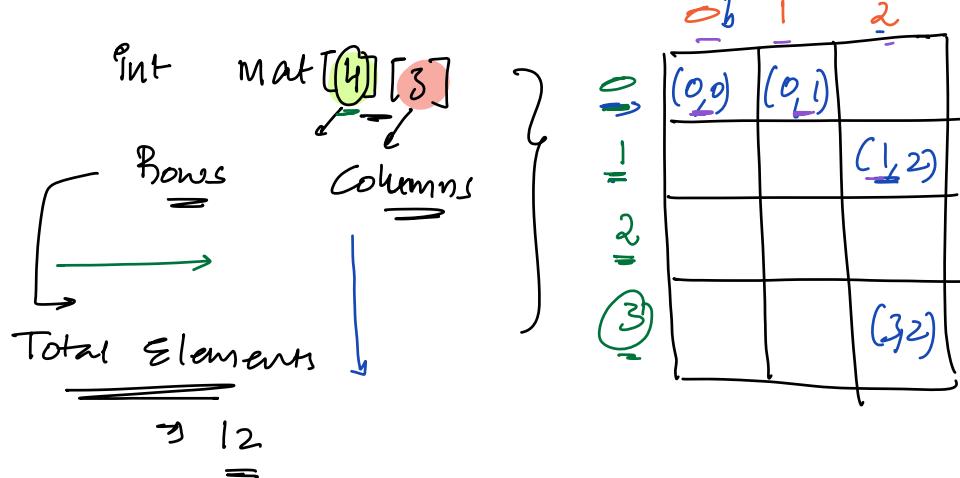
- Bits Manipulations
- Time Complexity
- Subarrays
- Prefix Sum
- Carry forward
- Sliding Window
- Contribution Technique
- 2D
- Interview Problems

Maths  
Sorting  
HashMaps  
String  
String + HashMap  
Recursion  
Subset / Subsequence

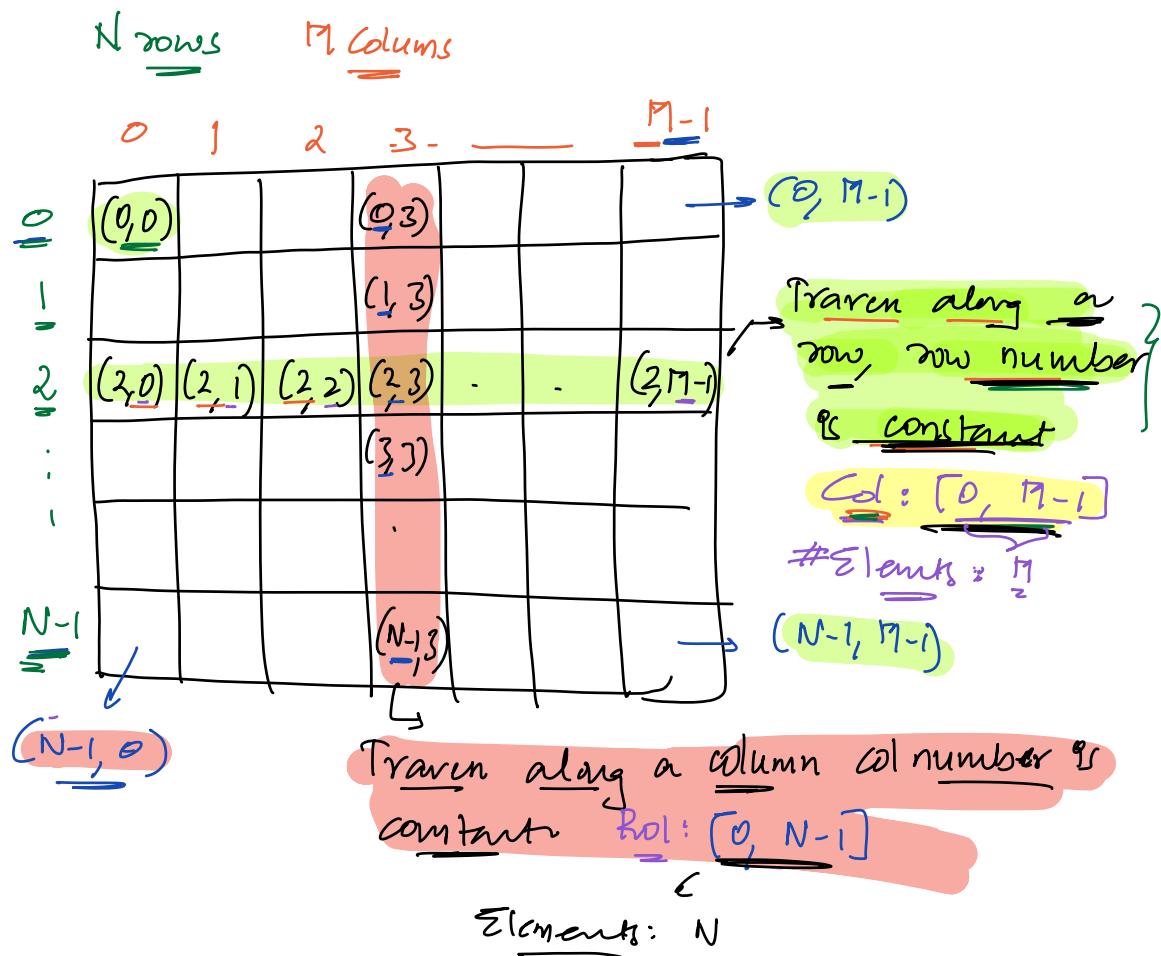
---

//  $\text{int arr}[5]:$

0	1	2	3	4



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



102 // Print row with sums

Given  $\text{Mat}[N][M]$ , print  $k^{\text{th}}$  row's sum

$[0 \dots N-1]$

$k = 0; k < N; k++) \{$

Iterating  $k^{\text{th}}$  row

$\underline{\text{sum} = 0}$

$j = 0; j < M; j++) \{$

$\underline{\text{sum}} = \underline{\text{sum}} + \text{Mat}[k][j]$

print (sum)

// Sum of every row right

Ex:

	0	1	2	3	
0	-2	1	6	3	: 8
1	2	9	3	8	: 22
2	3	-1	3	4	

Output

$k = 0 : j = [0 \dots 3] : 8$

$k = 1 : j = [0 \dots 3] : 22$

$k = 2 : j = [0 \dots 3] : 9$

// Print column wise sums.

	0	1	2	3
0	-2	1	6	3
1	2	9	3	8
2	3	-1	3	4

3    9    12    15

$$j=0 : r[0 \dots 2] = 3$$

$$j=1 : r[0 \dots 2] = 9$$

$$j=2 : r[0 \dots 2] = 12$$

$$j=3 : r[0 \dots 2] = 15$$

$i = 0; i < m; i++ \{$

// get  $j^{\text{th}}$  column sum

sum = 0

$p = 0; p < n; p++ \{$

sum = sum + mat[i][p]

print(sum)

Q3) Given Square Matrix  $\{N \times N\}$ ,

$N=4$

	0	1	2	3
0	(0,0)	-	-	-
1	-	(1,1)	-	-
2	-	-	(2,2)	-
3	-	-	-	(3,3)

Print L-R diagonal

$$i=0; j=N; i++ \{$$

print mat[i][j]

TC:  $O(N)$

$N=4$

Print R-L diagonal

	0	1	2	3
0	-	-	-	-
1	-	-	-	-
2	-	-	-	-
3	-	-	-	-

Observations:

- 1)  $i$  is increasing by 1
- 2)  $j$  is decreasing by 1
- 3)  $i+j = N-1$

Mat  $[N][N]$

	0	1	.	.	$N-1$	$i$	$j$	$i+j$
0	-	-	-	-	-	(0, N-1)	= N-1	N-1
1	-	-	-	-	-	(1, N-2)	= N-1	N-1
2	-	-	-	-	-	(2, N-3)	= N-1	N-1
.	.	.	.	.	.	:		
$N-1$	-	-	-	-	-	(N-1, 0)	= N-1	N-1

$i$  is increasing by 1

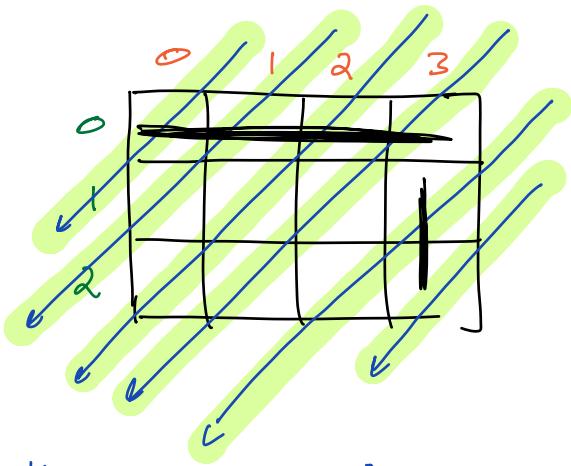
$j$  is decreasing by 1

// Printing R-L Diagonal Square Matrix

```
i = 0, j = N-1
while (i < N && j >= 0)
    print(Mat[i][j])
    i++; j--
}
TC: O(N)
```

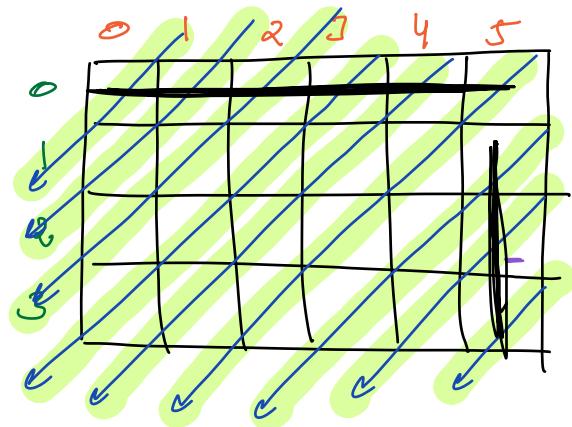
Q2) Given Rectangular Matrix, Print all Diagonals going from R-L.

mat[3][4]



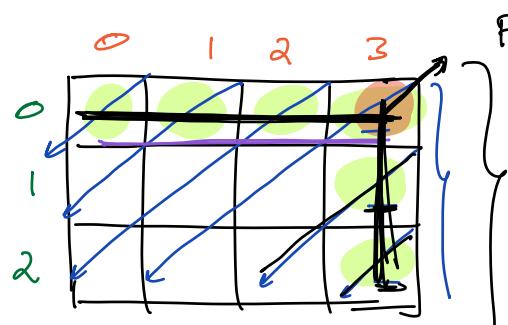
# How many diagonals R-L : 6

mat[4][6]



# How many diagonals from R-L = 9

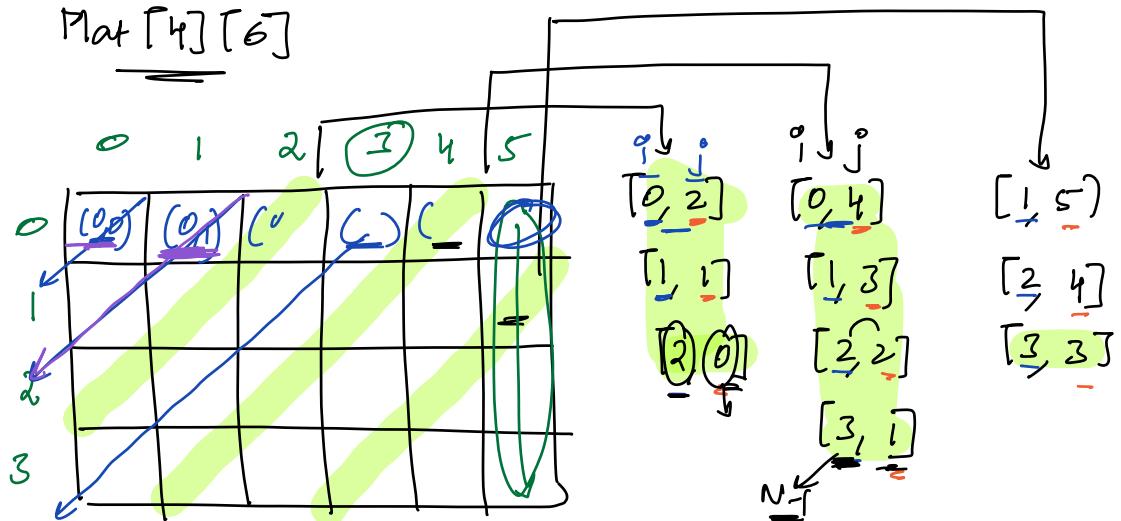
# How many diagonals from R-L = N+M-1



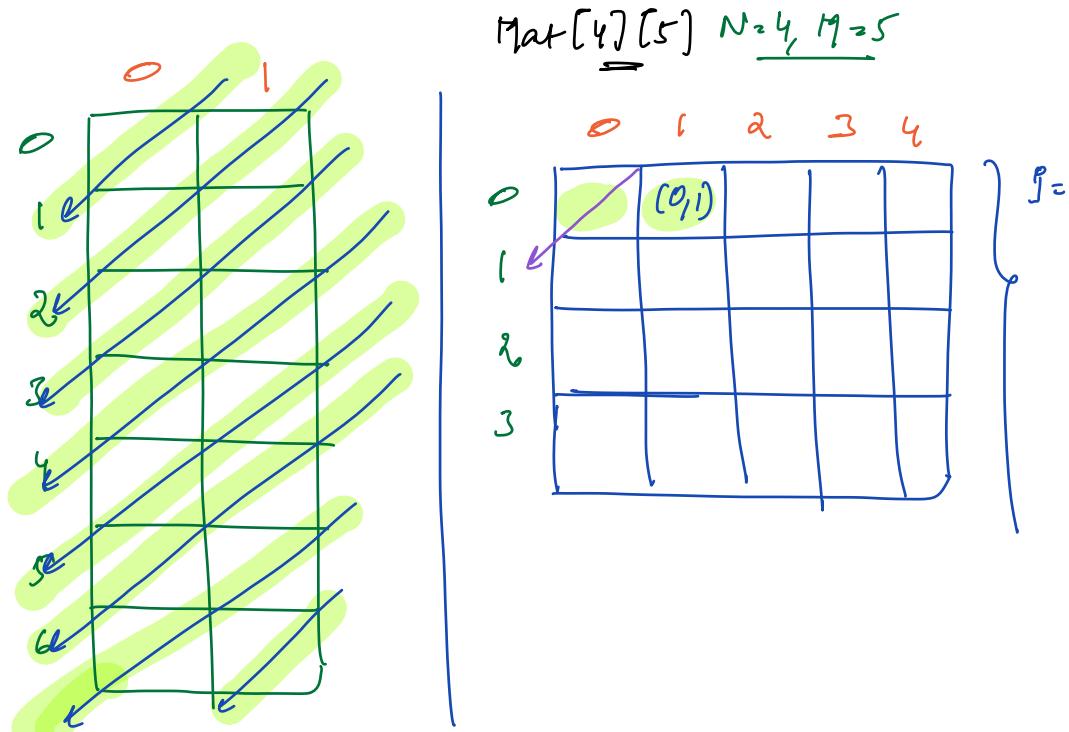
From which cells our diagonals can start  
# cells

- From Any cell in 0<sup>th</sup> Row : 3
- From Any cell in M-1<sup>th</sup> Column : N
- Total Diagonals  $\Rightarrow \frac{N+M-1}{2}$  ↗ top right corner twice

// Putting all diagonals from R-L



// Across any diagonal  $R-L$ , after each cut,  $i+1, j-1$



// Right diagonals can start from 0<sup>th</sup> row a last column  
 $\underline{C_{M-1}}$

→

```

    i = 0; j < M; i++ {
        (p = 0, j)
        while (i < N && j >= 0) {
            point(mat[i][j])
            p++; j--
        }
    }
}

```

// Pointing all diagonals  
starting in 0<sup>th</sup> row

$i = 1; i < N; i++ \rightarrow$  to avoid repetition.

```

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

```

Pointing all diagonals  
starting in M-1 column

$\Theta(N^2)$

## Transpose:

Mat [3][4]

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

Transpose

-3	8	2
6	7	1
2	3	6
1	4	4

0<sup>th</sup> Row  $\rightarrow$  0<sup>M</sup> Col

1<sup>st</sup> Row  $\rightarrow$  1<sup>M</sup> Col

i<sup>th</sup> Row  $\rightarrow$  i<sup>M</sup> Col

Rows = 4      Columns = 3

$\Rightarrow \underline{(2, 5)}$

	0	1	2	3	4
0	-6	3	9	14	29
1	2	8	11	16	15

Transpose

-6	2
3	8
9	11
14	16
29	15

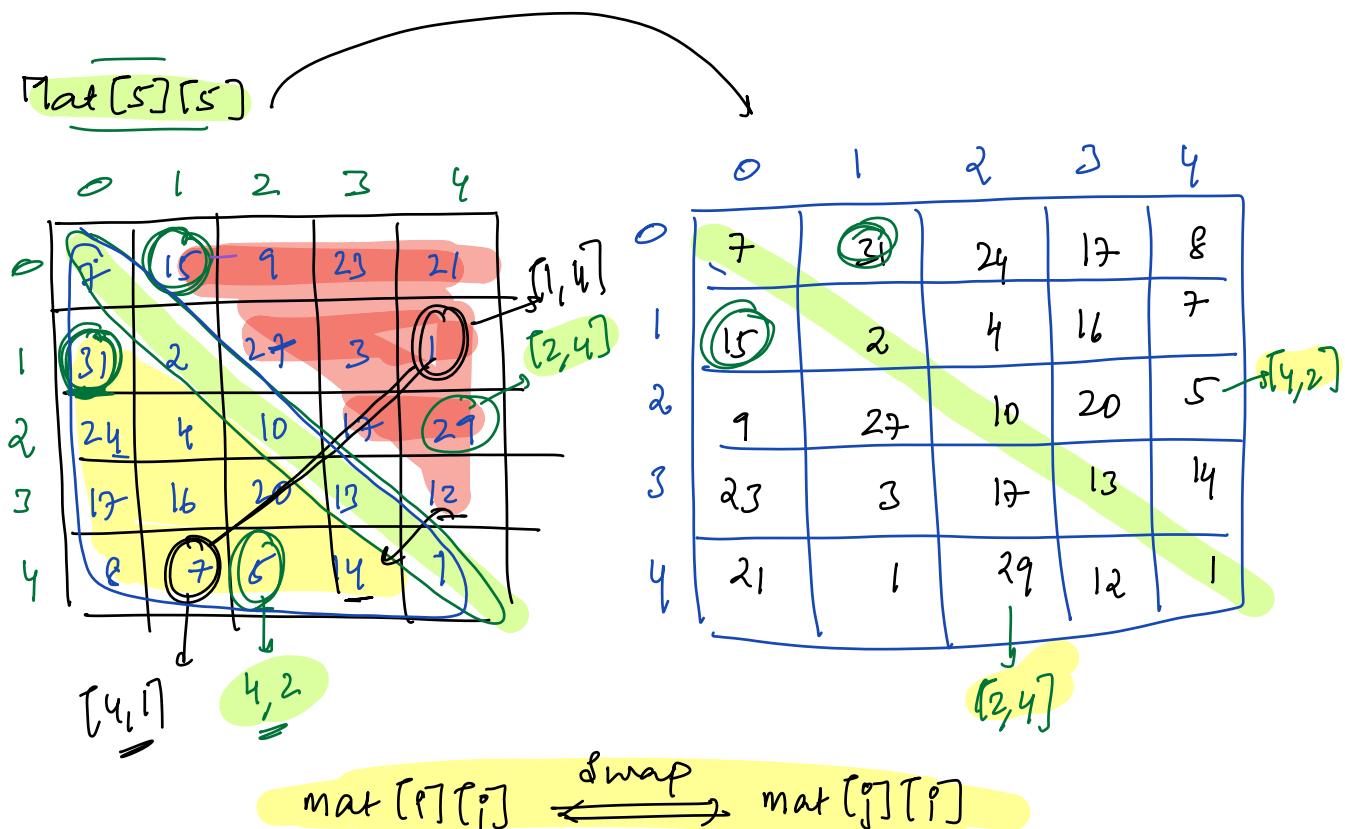
$\Rightarrow \underline{(5, 2)}$

// Q: Mat [N][M]  $\xrightarrow{\text{Transpose}}$  Mat [M][N]

// Q: Mat [N][N]  $\xrightarrow{\text{Transpose}}$  Mat [N][N] }  $\hookrightarrow$  diag Not changing

// Given a Square Matrix perform Transpose of Matrix inplace

{without extra  
space}



// Pseudocode

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

```
    j = 0; j < N; j++) {
```

swap(mat[i][j], mat[j][i])

when  $(i, j) = (0, 3)$

// Swap mat[0, 3] & mat[3, 0]

when  $(i, j) = (3, 0)$

// Swap mat[3, 0] & mat[0, 3]

Same matrix

// Either iterate in upper Triangle or lower Triangle  
TODO

→ let's iterate lower Triangle & swap

	0	1	2	3	4
0	(0,0)				
1	(1,0)	(1,1)			
2	(2,0)	(2,1)	(2,2)		
3	(3,0)	(3,1)	(3,2)	(3,3)	
4	(4,0)	(4,1)	(4,2)	(4,3)	(4,4)

$i = 0; i \leq N; i++ \{$   
 $j = 0; j \leq i; j++ \}$   
 swap mat[i][j] &  
 mat[j][i]

→ TC:  $O(N^2)$  SC:  $O(1)$  →  $\underline{\underline{O(N^2)}}$

// Doubts

$i = 0; i \leq N/2; i++ \{$   
 // Swap mat[i][N-i-1]  
 $mat[N-i-1][i]$

$N = 5$   
 $i \quad N-i-1 \quad N-i-1 \quad 0$   
 $i=0: mat[0][4] - mat[4][0]$   
 $i=1: mat[1][3] - mat[3][1]$   
 $i=2: mat[2][2] - mat[2][2]$   
 $i=3: mat[3][1] - mat[1][3]$   
 $i=4: mat[4][0] - mat[0][4]$

// Given  $\text{mat}[N \times N]$  rotate by  $90^\circ$ , from **clockwise**

$$\underline{N=5}$$

$\left\{ \begin{array}{l} \rightarrow \text{Step 1: Transpose mat} \\ \rightarrow \text{Step 2: Reverse each row} \end{array} \right.$

	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

Rotate  $90^\circ$  clockwise

Transpon

or

	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 0<sup>th</sup> row

1<sup>st</sup> row

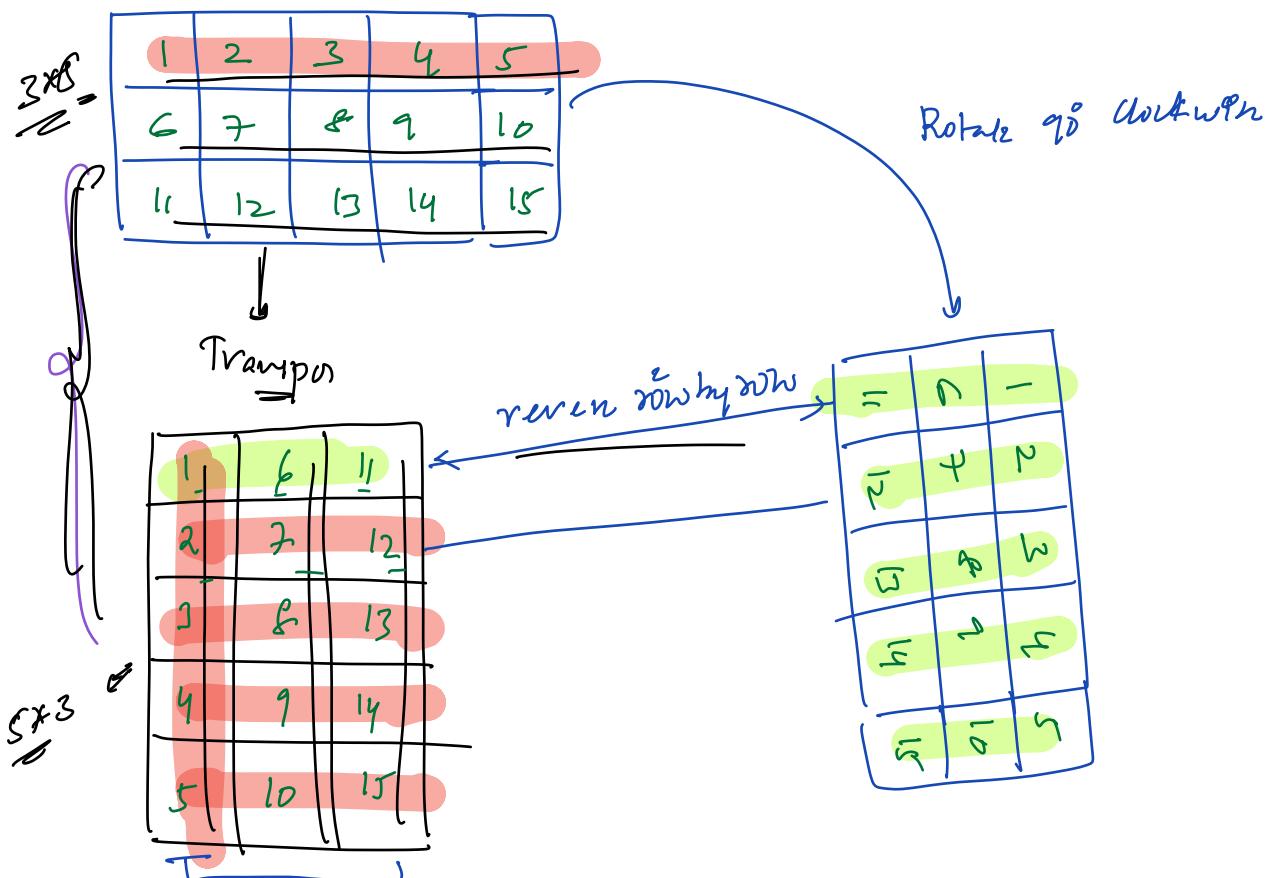
2<sup>nd</sup> row

3<sup>rd</sup> row

4<sup>th</sup> row

	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  $90^\circ$  Rectangular



→ For a rectangular you need extra space to calculate transpose of a matrix.

→