

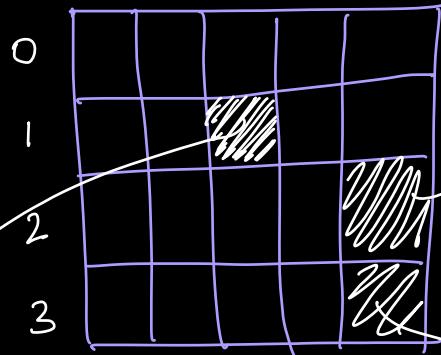
$\text{mat}[4][5]$

rows

columns

$\text{mat}[1][2]$

0 1 2 3 4



$\text{mat}[2][4]$

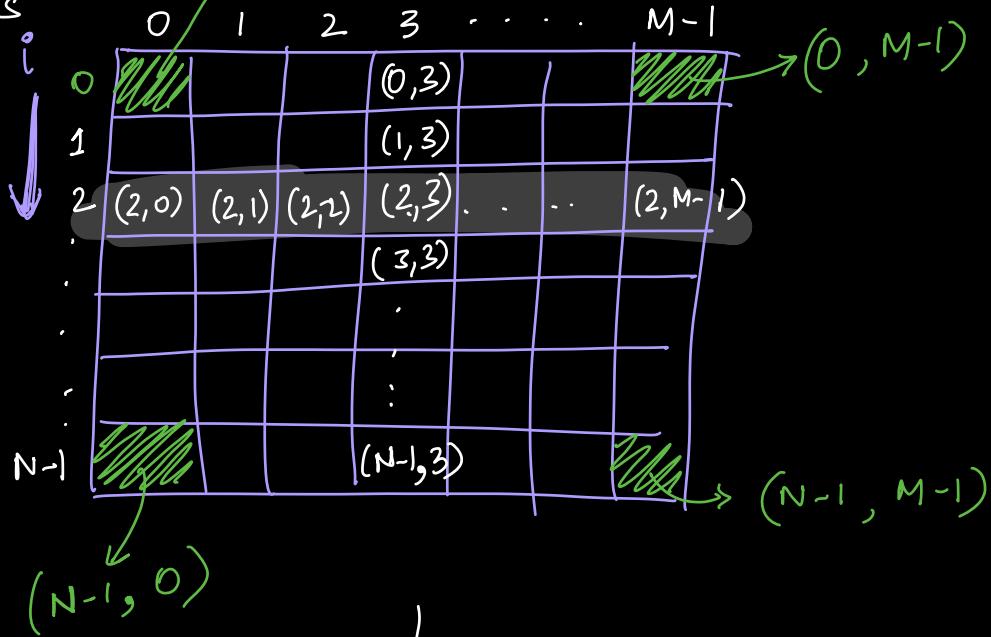
$\text{mat}[3][4]$

(N)  $\leftarrow$  rows

(M)  $\leftarrow$  columns

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

$j \rightarrow (0, 0)$



Traverse a row

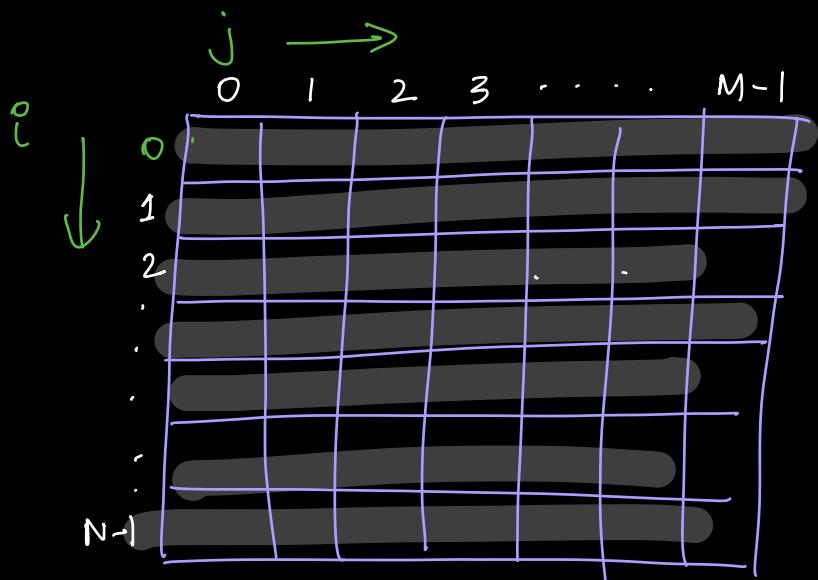
$\rightarrow$  Row no. will be constant

$\rightarrow$  Col  $\rightarrow [0, M-1]$

COL WISE

$\rightarrow$  Col no. will be constant.

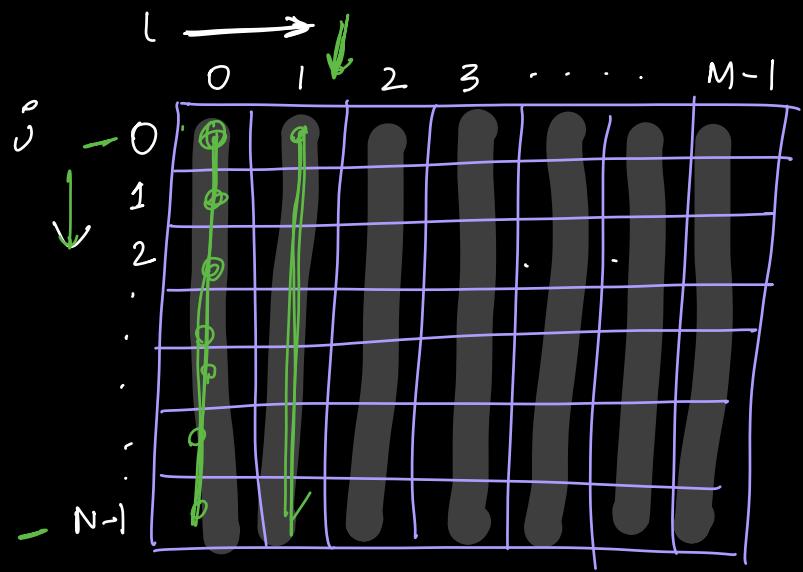
$\rightarrow$  Row  $\rightarrow [0, N-1]$



```

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

```

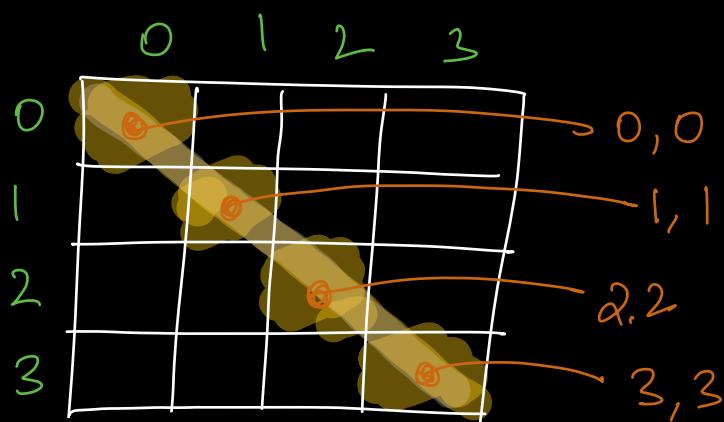
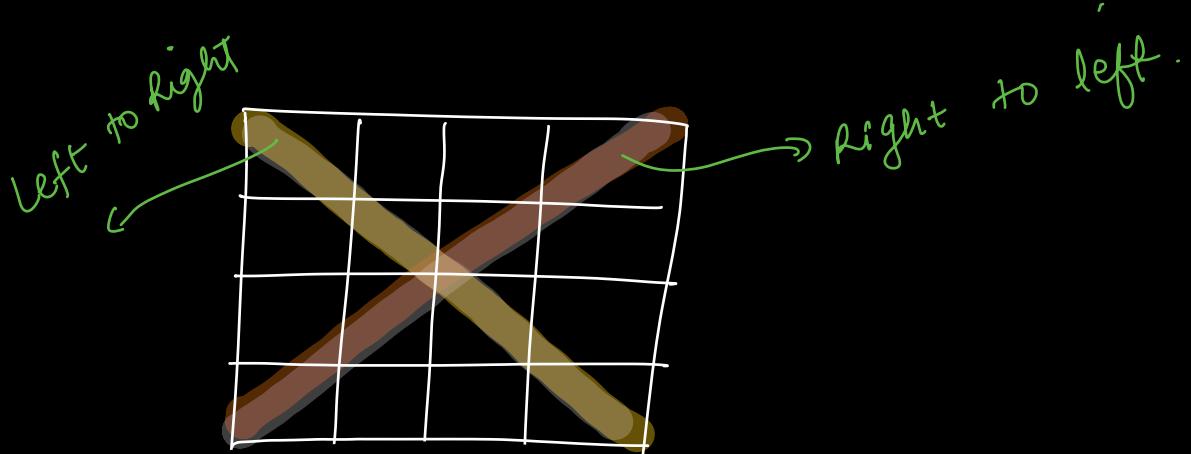


```

for (i = 0; i < M; i++) { i = 1
    sum = 0
    for (j = 0; j < N; j++) {
        |   sum + = mat [j][i]
    }
    print (sum)
}

```

Q Given a square matrix,  $N=M$ ,



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

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

```
        if (i == j) {
```

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

$O(N^2)$

}

}

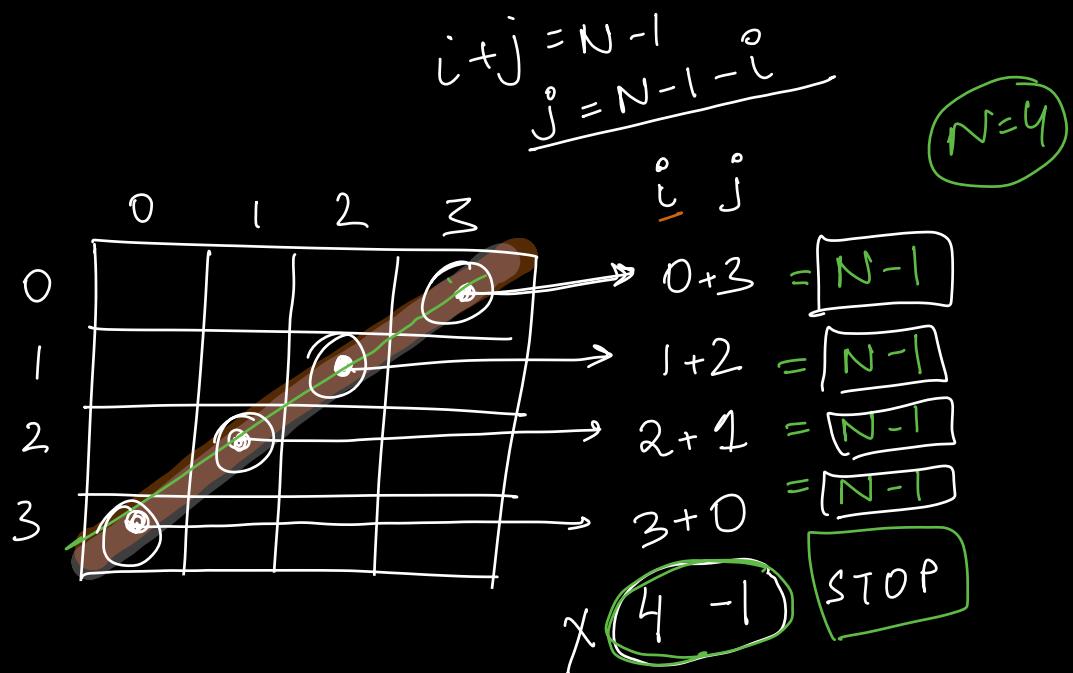
```

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

```

$\frac{TC}{O(N)}$

$\left\{ \begin{array}{l} mat[0][0] \\ mat[1][1] \\ mat[2][2] \\ mat[3][3] \end{array} \right.$



```

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

```

$O(N)$

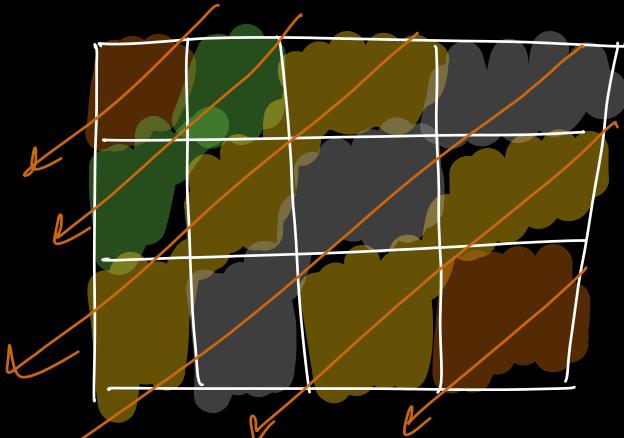
$\left\{ \begin{array}{l} \text{for}(i=0; i<N; i++) \\ \quad j = N-1-i \\ \quad \text{print}(mat[i][j]) \end{array} \right.$

$O(N)$

Ques. Given a rectangular matrix, print all the diagonals going from right to left.

$$N=3 \quad M=4$$

$\text{mat}[3][4]$

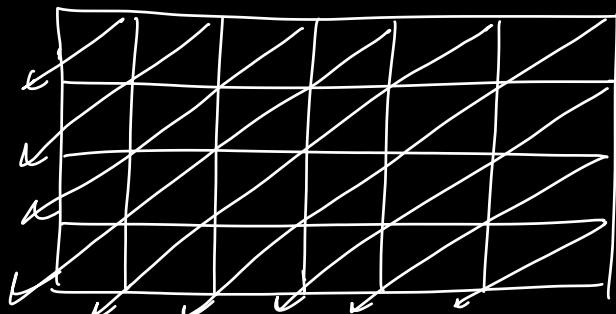


Right to left.  
With single diagonal

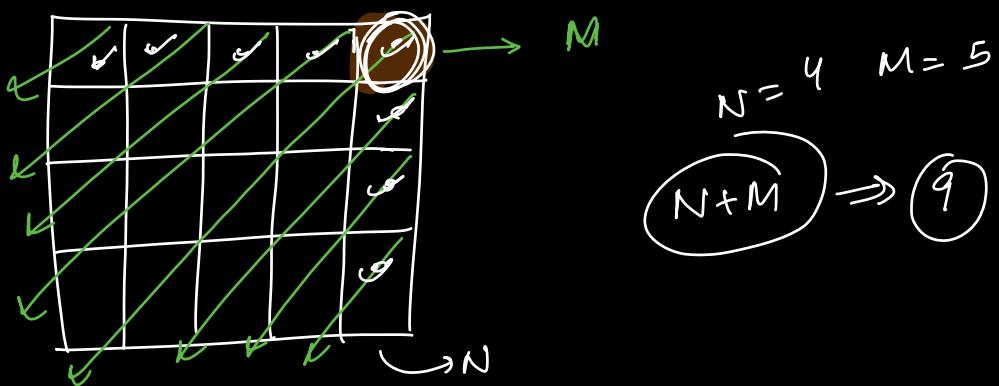
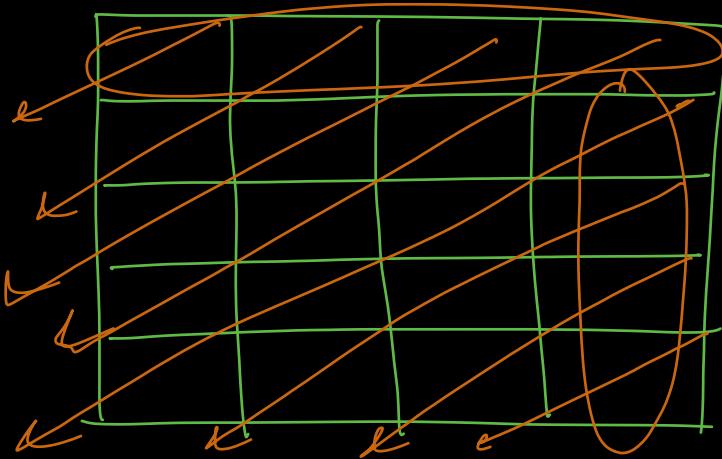
4	6	2	1		
5	3	1	9		
2	7	8	10		

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

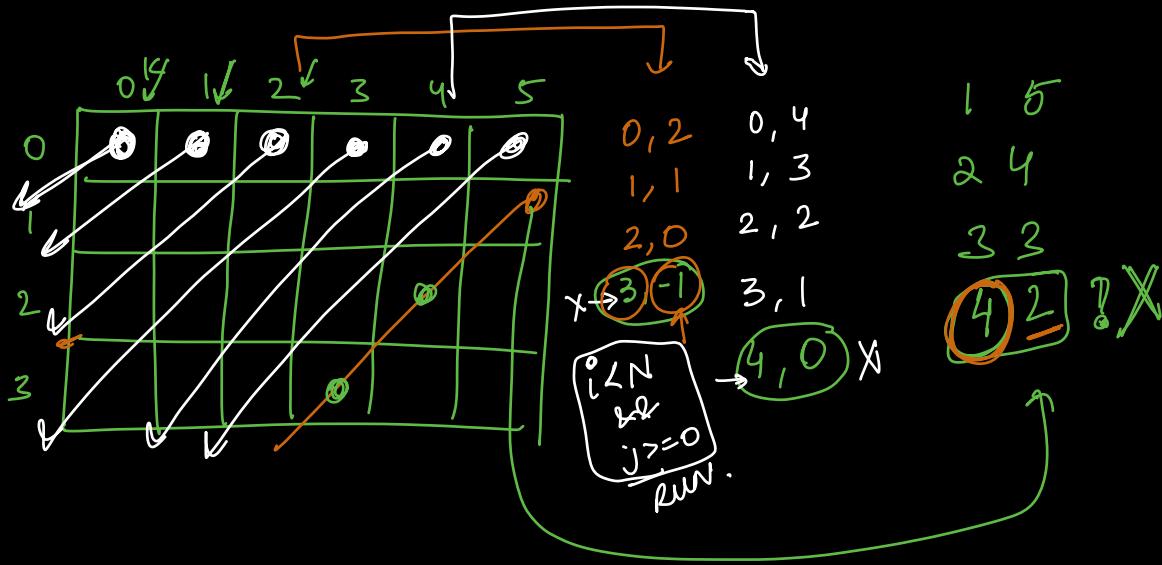
$\text{mat}[4][6]$



⑨ diagonals.



Total diagonals are  $\rightarrow N + M - 1$



====  
all points  
on diagonals  
originating  
from first  
row

====

$$K = 0$$

$$\begin{matrix} i = 0 & j = 0 \\ | & -| \end{matrix}$$

2

$$K = 1$$

$$\begin{matrix} i = 0 & j = 1 \\ | & 0 \end{matrix}$$

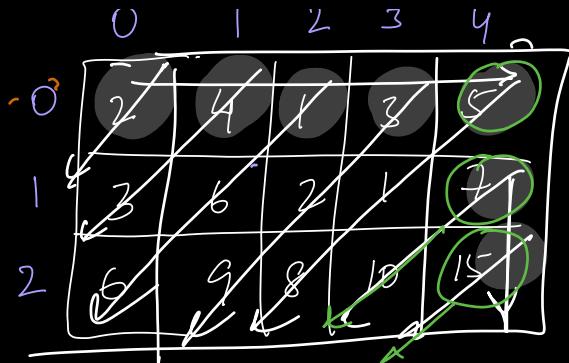
cells in  
last col  
 $\rightarrow N$

1 to  $N-1$

4, 3

$$K = 2$$

$$\begin{matrix} i = 0 & j = 2 \\ | & | \\ 2 & 0 \\ \hline 3 & -1 \\ \hline 1 & 6 & 6 \end{matrix}$$



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

$i = 0 \quad j = K$  // starting cells of each diagonal.  
 while ( $i \leq N$ ) {  
 $j = K$ ;

White RN 4\*

print(marci[j])

i++  
j--

1

The diagram illustrates a stack-based insertion algorithm for a linked list. It shows a stack frame with local variables  $i$  and  $j$ , both initialized to 0. The stack grows downwards. A condition box at the top checks if  $i < N$  and  $j \geq 0$ . If true, it performs an insertion operation, which is shown as a box with  $i = 0$  and  $j = 1$ . This leads to three possible states:

- $i = 0, j = 1$  (current state)
- $i = 0, j = 2$  (next state after insertion)
- $i = 0, j = 3$  (next state after another insertion)
- $i = 0, j = 4$  (final state after the last insertion)

On the right, the stack grows downwards from top to bottom, with values 2, 4, 3, 1, 6, 5, 2, 9, and 5, 1, 8. The label "N-1" is written vertically on the left side.

for( K = 1 ; K < N ; K++ ) {

$$i = K \quad j = M - 1$$

$\{ i < n \wedge k^i j = 0 \}$

point( mat[i][j] )  
i++  
j = -

2

Q. Given a square matrix, perform transpose of matrix

mat [5][5]				
	0	1	2	3
0	29	28	8	23
1	10	7	3	17
2	19	2	24	5
3	10	14	96	33
4	30	0	98	13
				20


Transpose

$(3,1)$  Diff. matrix

Time :  $O(N \times M)$   
Space :  $O(N \times M)$

$\backslash$   
 $(3,1)$

$i, j \leftrightarrow j, i$

DO NOT USE EXTRA SPACE ↴

$i, j \xrightarrow{\text{transpose}} \text{swap is happening}$   
 $i, j \leftrightarrow j, i$

{ for( $i=0; i < N; i++$ ) {  
|     { for( $j=0; j < M; j++$ ) {  
|     |     Swap( $\underline{\underline{\text{mat}[i][j]}}$ ,  $\underline{\underline{\text{mat}[j][i]}}$ )  
|     }  
}  
} }  
 $i = 0 \quad j = 3 \quad \text{swapped with } (3,0)$

$i = 3 \quad j = 0 \quad \text{swapped } (0,3)$

$i$	$j$	0	1	2	3	4
0	29	28	8	23	30	
1	10	7	3	11	17	
2	19	2	24	5	25	
3	10	14	96	33	18	
4	30	0	98	13	20	

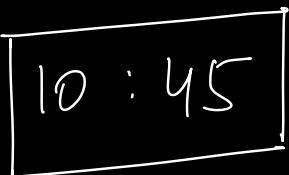
$$\begin{aligned}
 i &= 0 & j &= [0 \quad 0] \\
 i &= 1 & j &= [0 \quad 1] \\
 i &= 2 & j &= [0 \quad 2] \\
 i &= 3 & j &= [0 \quad 3] \\
 &\vdots && \vdots \\
 i &= N-1 & j &= [0 \quad N-1]
 \end{aligned}$$

```

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

```

$$\begin{aligned}
 \text{temp} &= a \\
 a &= b \\
 b &= \text{temp}
 \end{aligned}$$

Break : 

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

Note:  
Rectangular matrix, you'll need extra matrix  
to find the transpose

Ques. Given a square matrix, rotate the matrix from clockwise by 90°

American  
Cisco  
Adobe  
Uber  
Google  
Apple  
FB.

	0	1	2	3	4
0	29	28	8	23	30
1	10	7	3	11	17
2	19	2	24	5	25
3	10	14	96	33	18
4	30	0	98	13	20

rotate

	0	1	2	3	4
0	50	10	19	10	29
1	0	14	2	7	28
2	98	96	24	3	8
3	13	33	5	11	23
4	20	18	25	17	30

Transpose

1

↓

29	10	19	10	30
28	7	2	14	0
8	3	24	96	98
23	11	5	33	13
30	17	25	18	20

Total  
T.C =  $O(N \times M)$  +  $O(1)$  S.C

=  $O(N \times M)$

S.C =  $O(1)$

① Transpose the given matrix

② Reverse each row.

$O(N \times M)$   
 $O(1)$

If rectangular matrices

T.C  $\rightarrow O(N \times M)$   
S.C  $\rightarrow O(N \times M)$

Next session  $\rightarrow$  Interview Problems

Double

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

$N \times M$

Rectangular

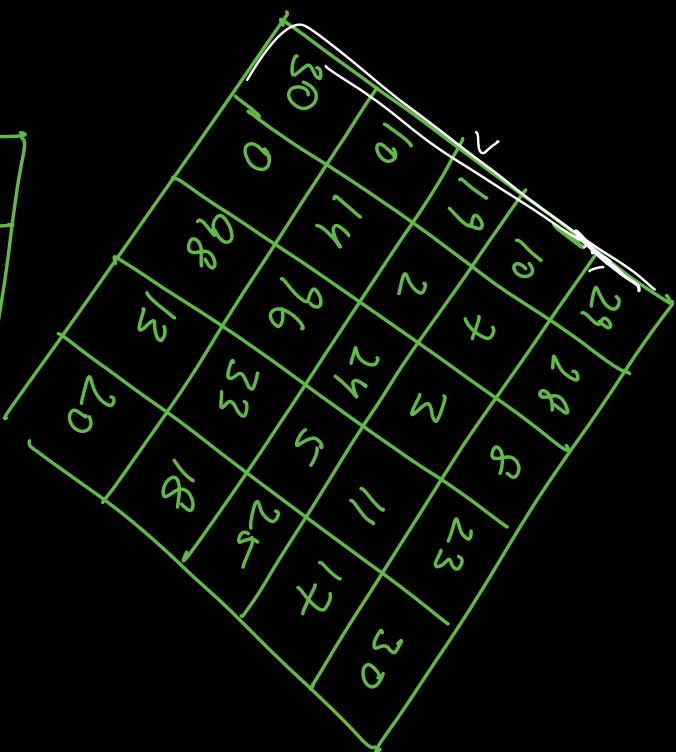
III.

extra matrix

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

$M \times N$

	0	1	2	3	4
0	29	28	8	23	30
1	10	7	3	11	17
2	19	2	24	5	25
3	10	14	96	33	18
4	30	0	98	13	20



	0	1	2	3	4
0	29	28	8	23	30
1	10	7	3	11	17
2	19	2	24	5	25
3	10	14	96	33	18
4	30	0	98	13	20

0	20	13	86	0	30	h
1	18	33	96	11	01	s
2	25	5	24	2	11	a
3	17	11	2	10	01	i
4	29	28	8	23	0	o
5	30	1	7	13	20	

- ① Reverse col wise
- ② Reverse row wise.