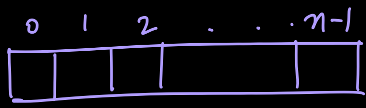
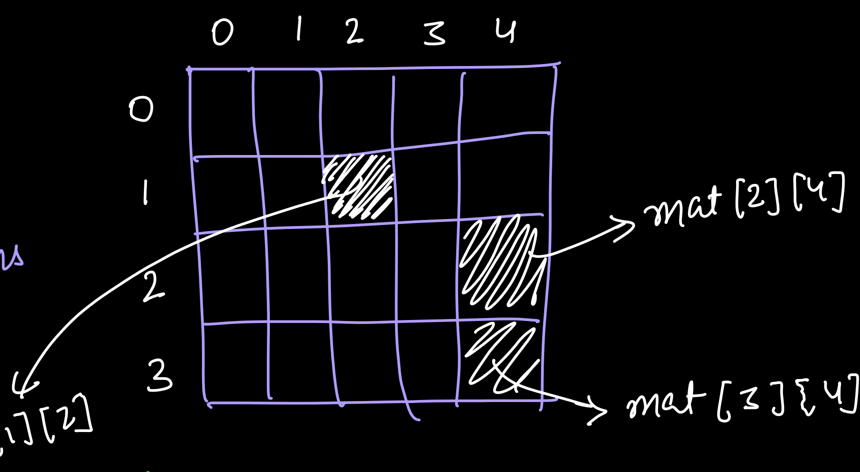



mat [4] [5]

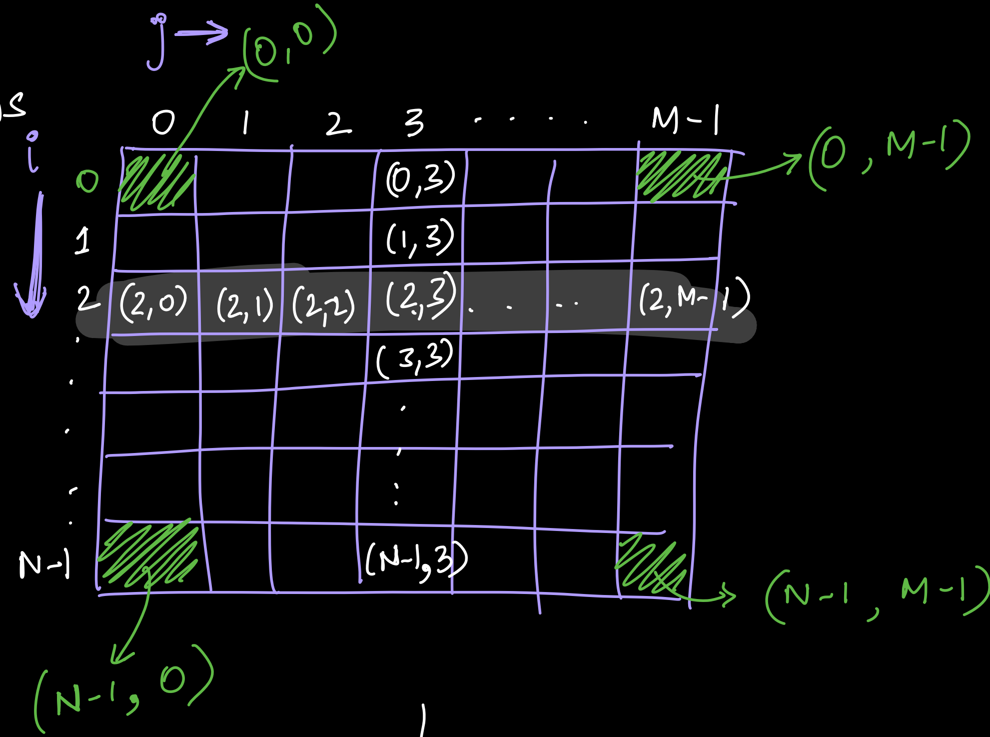
↓ rows ↓ columns




N ← rows

M ← columns

mat [N] [M]




Traverse a row

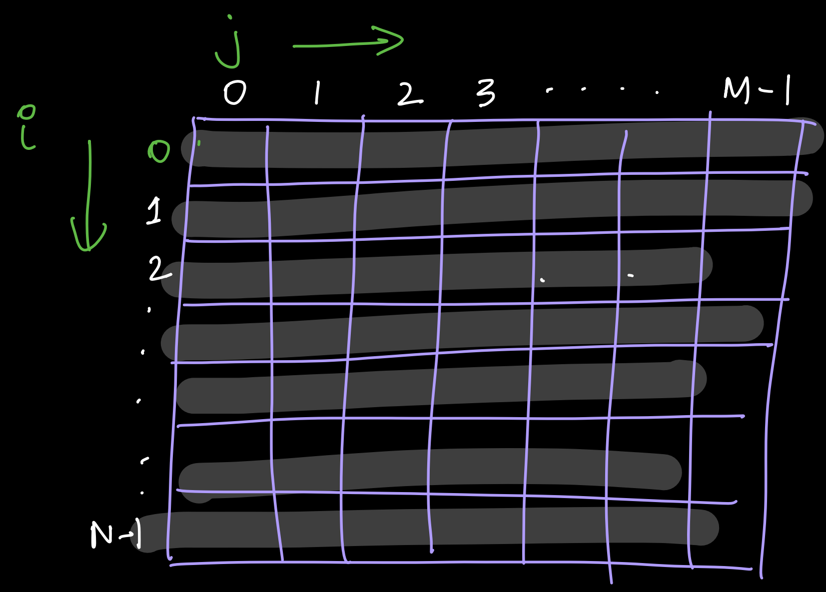
→ Row no. will be constant

→ col → [0, M-1]

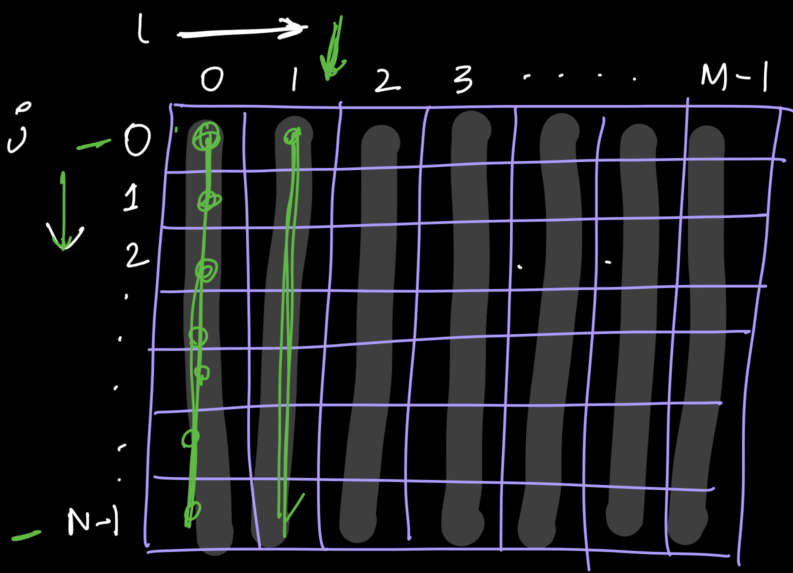
COL WISE

→ col no. will be const.

→ ROW → [0, N-1]



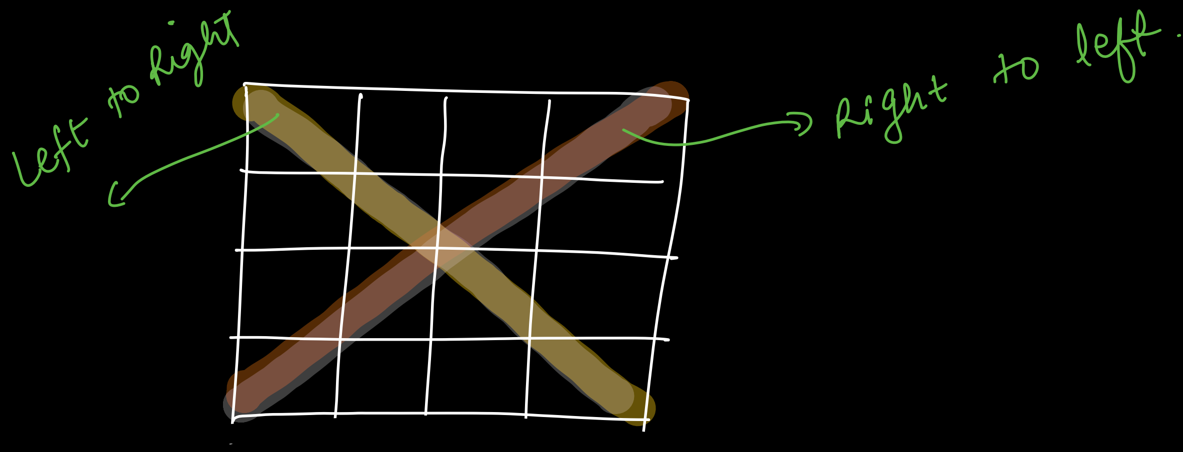

```
for(i=0; i<N; i++){
    sum = 0
    for(j=0; j<M; j++){
        sum += mat[i][j]
    }
    print(sum)
}
```

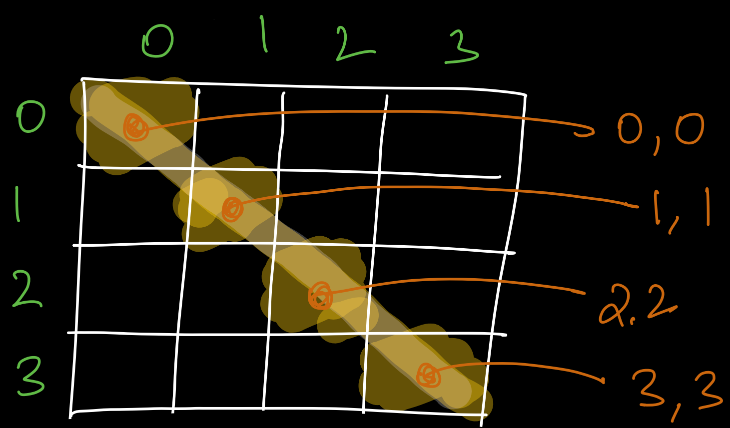



i=1

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

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




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

$O(N^2)$

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

TC

O(N)

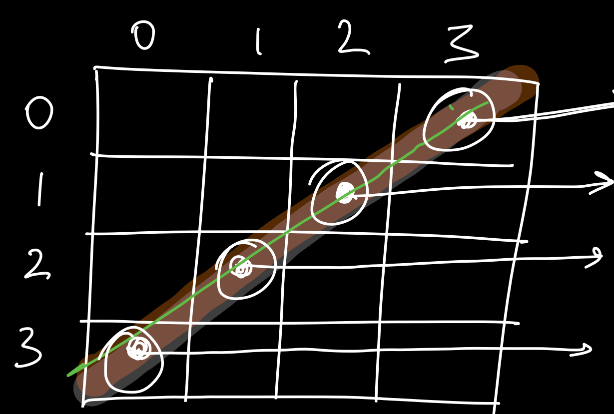
mat[0][0]
mat[1][1]
mat[2][2]
mat[3][3]

$i+j = N-1$

$j = N-1-i$

N=4

i j



0+3 = $\boxed{N-1}$

1+2 = $\boxed{N-1}$

2+1 = $\boxed{N-1}$

3+0 = $\boxed{N-1}$

~~4 -1~~ STOP

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

O(N)

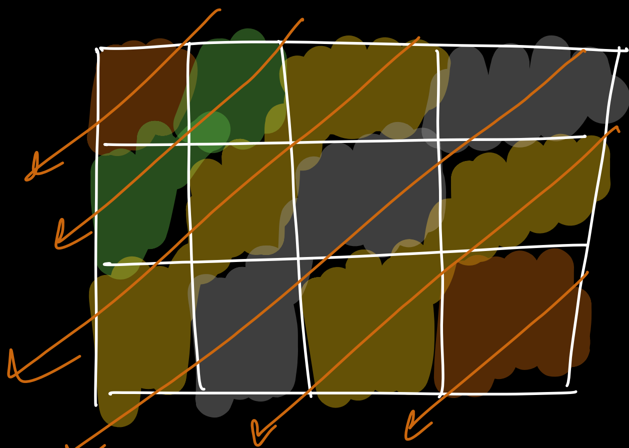
```
for(i=0; i<N; i++){
    j=N-1-i
    print(mat[i][j]
}
```

O(N)

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

N=3  M=4

mat[3][4]



Right to left.

w.r.t single diagonal

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

4

6, 5

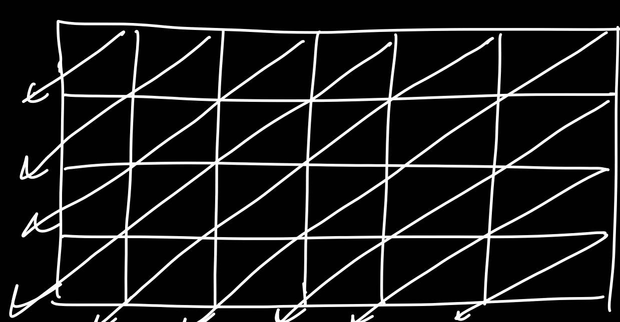
2, 3, 2

1, 1, 7

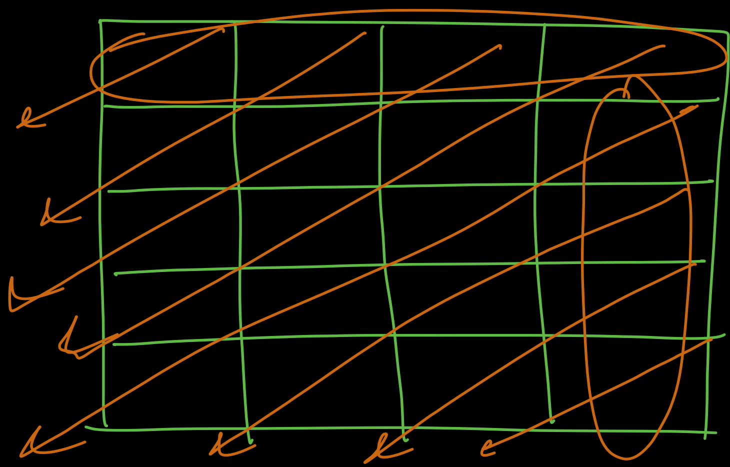
9, 8

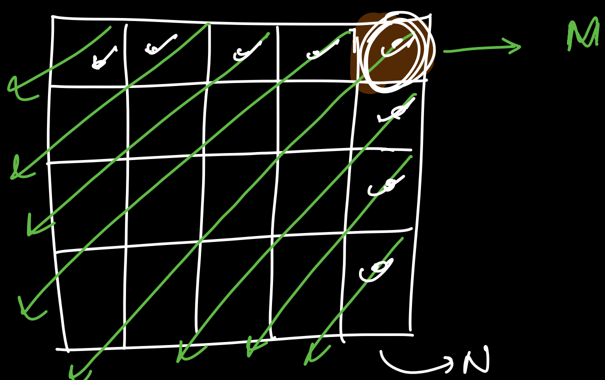
10

mat[4][6]



9 Diagonals.

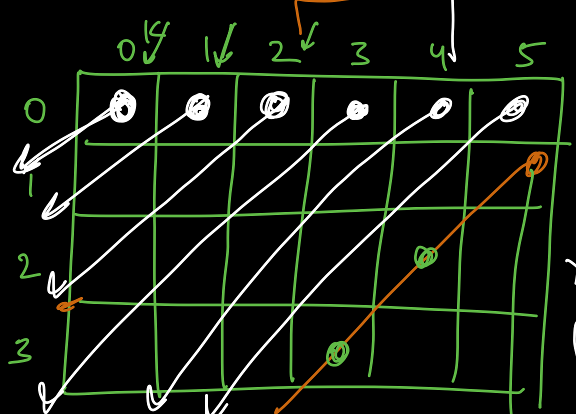





N = 4 M = 5

$N+M \Rightarrow 9$

Total diagonals are → $N+M-1$




| | | |
|---|---|---|
| 0, 2 | 0, 4 | 1 5 |
| 1, 1 | 1, 3 | 2 4 |
| 2, 0 | 2, 2 | 3 3 |
| x → 3, -1 | 3, 1 | 4 2 ?X |
| | 4, 0 X | |

$i<N$ && $j>=0$ RUN.

prints all diagonals originating from first row

K = 0

i = 0 j = 0

1 -1

2

K = 1

i = 0 j = 1

1 0

4, 3

cells in last col → N

1 to N-1

K = 2

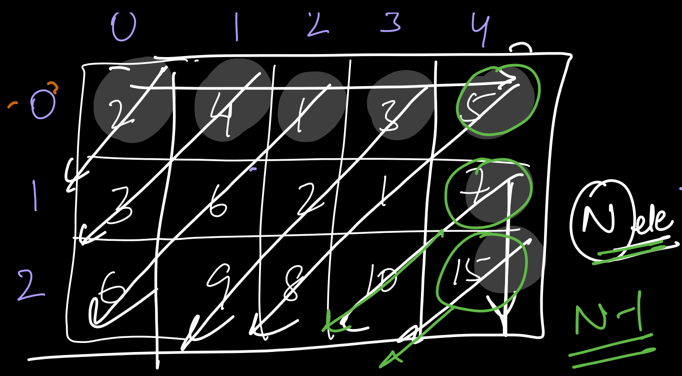
i = 0 j = 2

1 1

2 0

3 -1

1 6 6




$i=0$ $j=0$

$\boxed{i<N \quad j>=0}$

$i=0$ $j=1$

$i=0$ $j=2$

$i=0$ $j=3$

$i=0$ $j=4$

2

4 3

1 6 6

3 2 9

5 1 8

```
for(K=0; K<M; K++){
    i=0   j=K      // starting cells of each
                      diagonal.
    while(i<N &&
              j>=0){
        print(mat[i][j])
        i++
        j--
    }
}
```

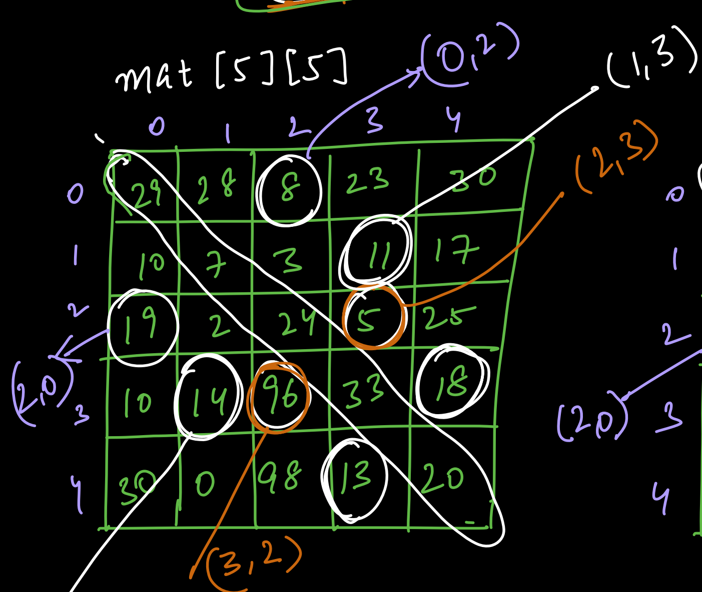
[1  N-1]

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

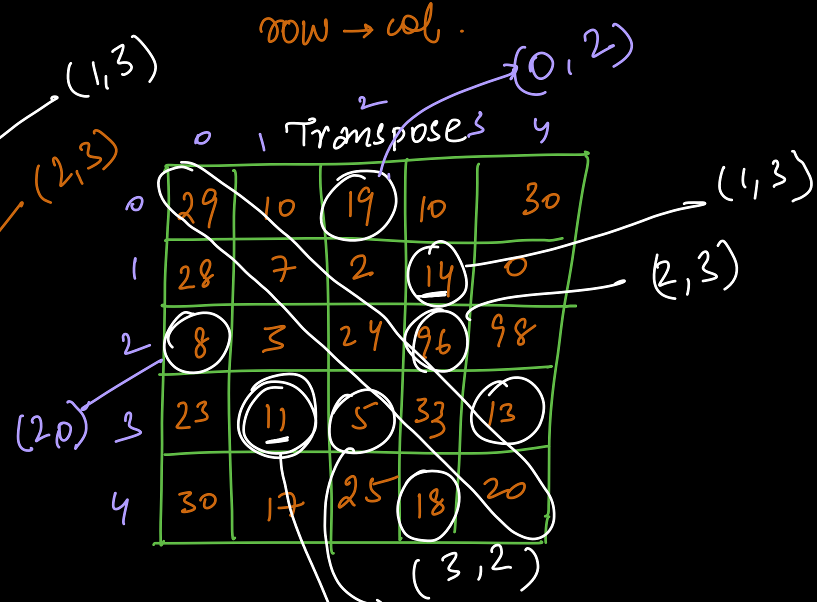
Q. Given a square matrix, perform transpose of matrix

row → col.

mat [5][5]




Transpose



(3,1) Diff. matrix

Time : O(N×M)

Space : O(N×M)

(3,1)

i,j ⟷ j,i

DO NOT USE EXTRA SPACE !

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

i,j ⟷ j,i

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

i = 0   j = 3   swapped with (3,0)

i = 3   j = 0   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 |

i = 0 j = [0 0]

i = 1 j = [0 1]

i = 2 j = [0 2]

i = 3 j = [0 3]

⋮

i = N-1 j = [0 N-1]

i-1

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

→ a b

temp = a
a = b
b = temp

Break : $\boxed{10:45}$

| | 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) | (4,1) |

---

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

Amazon
Cisco
Adobe
Uber
Google
Apple
FB.

**Que.** Given a square matrix, rotate the matrix from Clockwise by 90°

| | 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 |

$\xrightarrow{\text{rotate}}$

| | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|
| 0 | 30 | 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.

| | | | | |
|---|---|---|---|---|
| 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 |

rotated

1. Transpose the given matrix

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

   $O(1)$ S.C

2. Reverse each row.

   $O(N \times M)$

   $O(1)$

Total

T.C → $O(N \times M) + O(N \times M) = O(N \times M)$

SC → $O(1)$

If Rectangular matrices

TC → $O(N \times M)$

SC → $O(N \times M)$

Next session → Interview Problems.

Double

Rectangular

|||

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

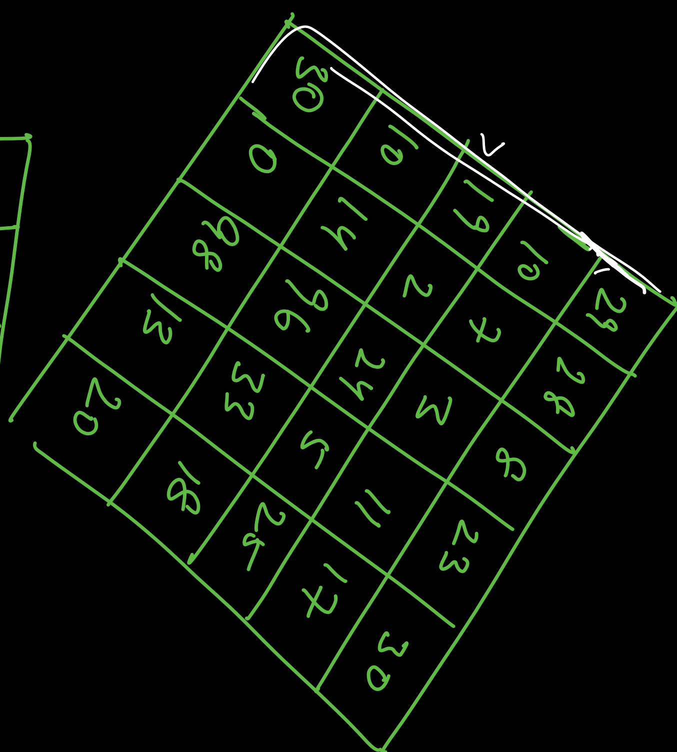
N × M

Extra matrix

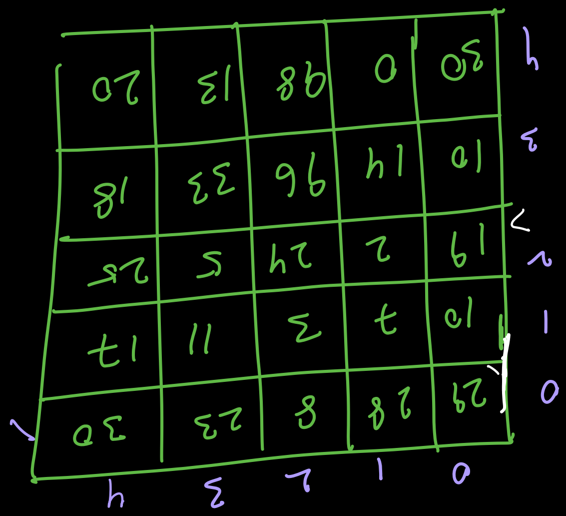
| | | |
|---|---|---|
| 4 | 1 | 2 |
| 6 | 5 | 3 |
| 9 | 7 | 10 |
| 8 | 1 | 5 |
| 9 | 5 | 3 |

M × 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 |



1. Reverse col wise
2. Reverse row wise.