

2D Matrices

Vinay Neekhra

Senior Instructor & Mentor

Reachable in Scaler Lounge 

"Computer Science is like a matrix;
it's all about how you manipulate the numbers"

Today's content
1) Basics
2) problems.

§. Declare

`int mat[4][5]`

rows, horizontal line

col, vertical line

	0	1	2	3	4
0					
1					
2					
3					

`mat[1][2]`

`mat[2][4]`

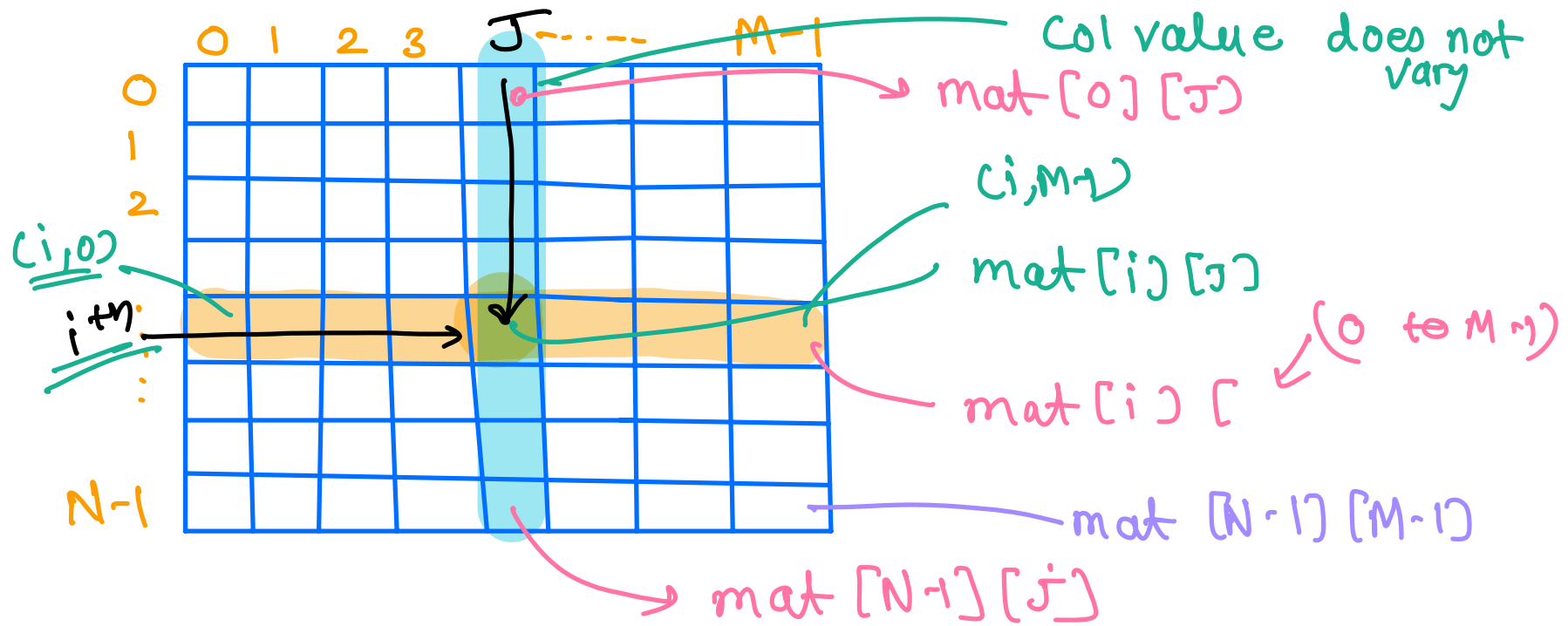
`int mat[N][M]`
N rows
M cols

row

`vector<int> row(M)`

`vector<vector<int>>`
`Mat(row, N)`

(C++ notation
you may ignore)



Obsⁿ:

- ① if we iterate on a row,
col no. changes from $[0, M-1]$
- ② if we iterate on a col,
row no. changes from $[0, N-1]$

Q Given $mat[N][M]$, print row wise sum.

ex.

	0	1	2	3	4	<u>Print</u>
0	2	-1	4	7	8	20
1	1	10	20	5	7	43
2	8	7	5	3	2	25
3	1	3	2	4	10	20

Quiz



find max number

T.C. $O(N)$

most optimal T.C.

void rowwisePrint (int mat[][]) {

int N = mat.length

int M = mat[0].length

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

sum = 0

// we need ith row's sum

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

print (sum)

T.C. = $O(N * M)$

S.C. = $O(1)$ ↑

most optimal

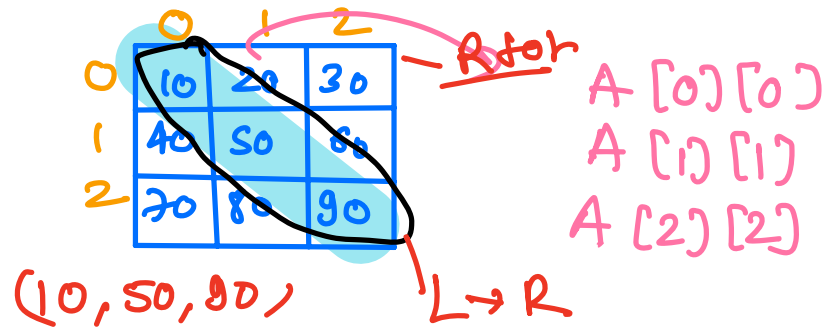
}

Q ex. Given mat [N][M], print col^m wise sum, ToDo
(can discuss during doubt session)

	0	1	2	3	4
0	2	-1	4	7	8
1	1	11	20	5	7
2	8	4	5	3	2
3	1	3	2	4	10

12, 17, 31, 19, 27

Q. Given a square matrix $A[N][N]$, print the diagonal

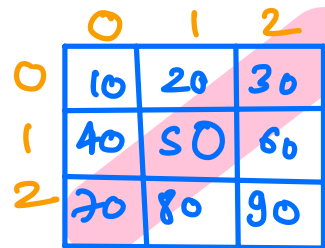


$L \rightarrow R$
 $(R-L)$

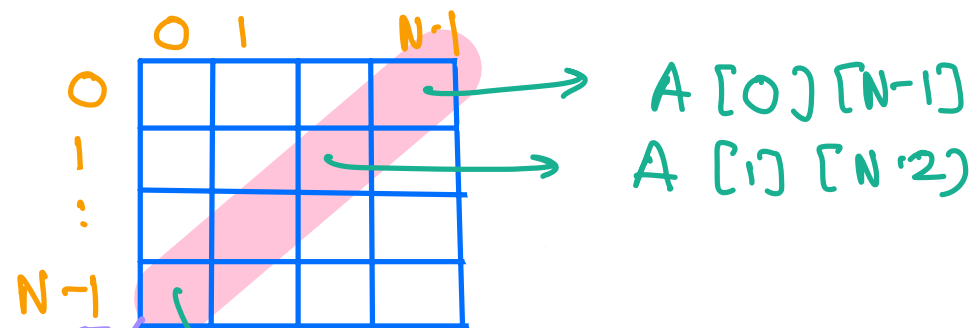
```
for (i=0; i<N; i++) {
    print A[i][i]
}
```

T.C. = $O(N)$
 S.C. = $O(1)$

Q. R to L Print diagonal elements.



Print (30, 50, 70)



obsn: ① row starts from 0 \rightarrow N-1 \uparrow
 $A[N][N-1] \times$

(2) col starts from $N-1 \rightarrow 0$ ↓

```
i = 0
j = N-1
while (i < N and j ≥ 0) {
    print A[i][j]
    i++;
    j--;
}
```

T.C. = $O(N)$
S.C. = $O(1)$

Q. Given a $A[N][M]$, print all the diagonals from R to L

ex.

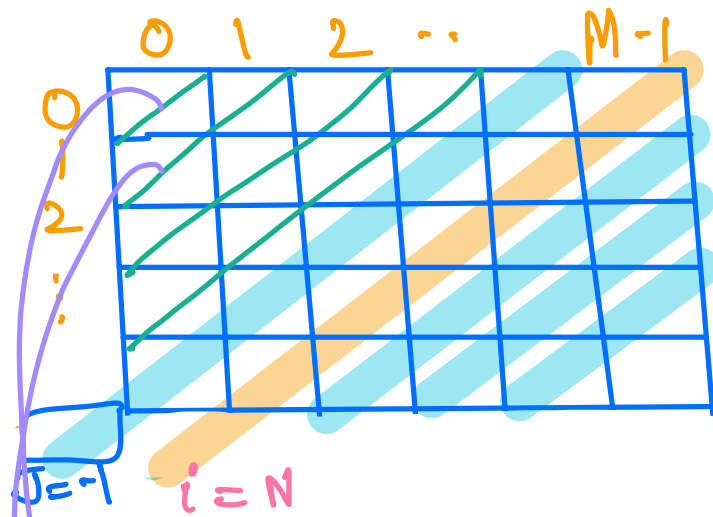
1st
2nd

	0	1	2	3	4
0	2	-1	4	7	8
1	1	11	20	5	7
2	8	4	5	3	2
3	1	3	2	4	10

ans.

Print

2					
-1	1				
4	11	8			
7	20	4	1		
8	5	5	3		
7	3	2			
2	4				
1	0				



obsⁿ $i \geq 0$ $J < M$
 (I) $i < N$ and $J \geq 0$

(II) when we are printing a diagonal
 J varies from $(0, \text{to } M-1)$
starting i was fixed at 0

(III) to print rest of the elements
 starting (J) is fixed $M-1$
 starting i will change

St. point $A[0][0]$

St. pt. 2nd diagonal $A[1][0]$

Solⁿ: (I) First print all the diagonal starting from
 0^{th} row

(II) print all the diagonals starting from
 last col^m.

starting index

(i)

(j)

for $(0, M-1)$

starting index i $[1, N-1]$ j $M-1$

```
void printAllDiagonals (int A[][]) {
```

```
    int N = A.length  
    int M = A[0].length
```

```
    for (c=0; c<M; c++) {  
        int i=0, j=c // starting index of the diagonal  
        while (i<N and j<=0) {  
            print (A[i][j])  
            i++;  
            j--;  
        }  
    }
```

11:05

```
    for (r=0r=1; r<N; r++) {  
        int j=M-1, i=r // starting index of the diagonal  
        while (i<N and j<=0) {
```



```

    print ( A[i][j] )
    i++;
    j--;
}
}

```

TC = $O(N \times M)$
 S.C = $O(1)$

	0	1	2	3	4
0	2	1	4	7	8
1	1	11	20	5	7
2	8	4	5	3	2
3	1	3	2	4	10

$N = 4, M = 5$

$C = 0 \rightarrow 2$

$i = 0 \rightarrow 1 \rightarrow 2 \rightarrow 3 \dots$

$j = 0 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \dots$

upper triangle

2
 1 1
 4 11 8
 7 20 4 1
 8 5 5 3

$N = 4, M = 5$
 $r = 2 \rightarrow 3$
 $j = 4 \rightarrow 3 \rightarrow 2 \rightarrow 1 \rightarrow 0$
 $i = 1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 2$

10:54 → 11:05

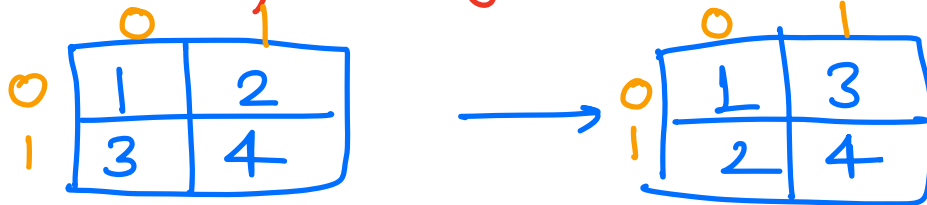
7 3 2

2 4

10

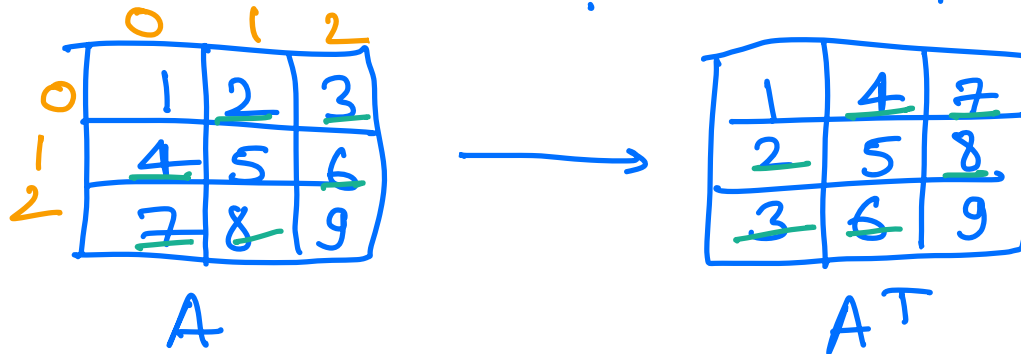
Q. Given a $A[N][N]$, calculate transpose of A .
with S.C. = $O(1)$

modify the given matrix to get the answer.



transpose →

0^{th} row $\Rightarrow 0^{th}$ col
 1^{st} row $\Rightarrow 1^{st}$ col
 \vdots



	0	1	2	3	4
0	2	-1	4	7	8
1	1	10	20	5	7
2	8	7	5	3	2
3	1	3	2	4	10
4	2	5	7	9	1

→

	0	1	2	3	4
0	2	1	8	1	2
1	-1	10	7	3	5
2	4	20	5	2	7
3	7	5	3	4	9
4	8	7	2	10	1

Solⁿ: swap every $A[i][j]$ with every $A[j][i]$

```
void transpose (int &A[][]) {
```

ignore this if you
are not comfortable
with calling
by address

```
int[][] transpose (int A[][]) {  
    // N, M  
    for (i=0; i<N; i++) {  
        for (j=0; j<M; j++) {  
            int temp = A[i][j]  
            A[i][j] = A[j][i]  
            A[j][i] = temp  
        }  
    }  
}
```

X won't work

$A[0][0] \leftrightarrow A[0][0]$

$A[3][3] \leftrightarrow A[3][3]$

1st solⁿ \Rightarrow visit
the lower
triangle
and swap

swapping

2nd solⁿ \rightarrow swap the upper
triangle
Hint

$A[i][i] = \text{temp}$

find the bug
& solve it

Diagram illustrating the transformation of a 2x2 matrix. The left matrix has elements 1, 2, 3, 4 with indices 0, 1 above and 0, 1 to the left. The element 1 is circled. An arrow points to the right matrix, which has elements 1, 32, 32, 4, with the first 32 crossed out.

Q. transpose of a matrix $A[N][M]$ (rectangle)

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

2x5

$T \rightarrow$

A 5x2 grid of numbers 1 through 10. The first column contains numbers 1 to 5, and the second column contains numbers 6 to 10. The grid is labeled "5x2" at the bottom right.

1	6
2	7
3	8
4	9
5	10

5x2

$$T.C = O(N^*M)$$

$$S.C = O(N^*M)$$

we need extra space

Q. Given a matrix $A[N][N]$, Rotate 90° clockwise

	0	1	2	3
0	1	2	3	4
1	5	6	7	8
2	9	10	11	12
3	13	14	15	16

Rotation
90°

	0	1	2	3
0	13	9	5	1
1	14	10	6	2
2	15	11	7	3
3	16	12	8	4

N
0
1

Matrix
[]
[1]

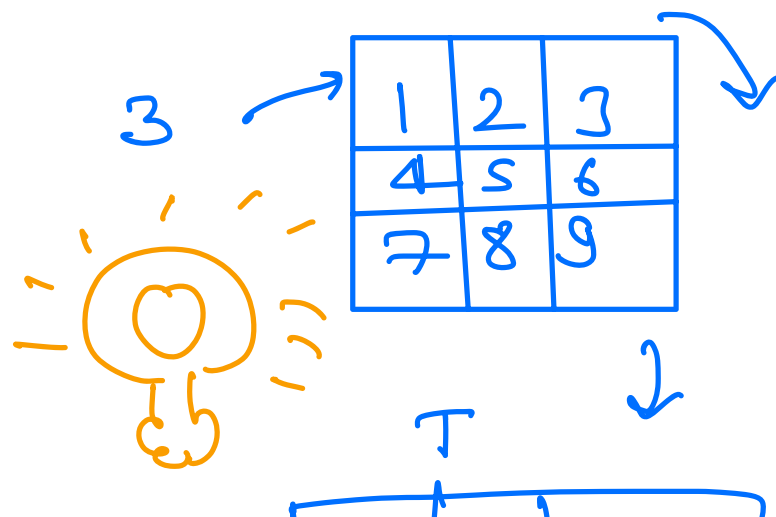
ans
[]
[1]

2

$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$

→

3 1
4 2



90°

7	4	1
8	5	2
9	6	3

(ans)

1	4	7
2	5	8
3	6	9

reversing each row

	0	1	2	3
0	1	2	3	4
1	5	6	7	8
2	9	10	11	12
3	13	14	15	16

go

	3	2	1	0
0	13	9	5	1
1	14	10	6	2
2	15	11	7	3
3	16	12	8	4

↓ T

1	5	9	13
2	6	10	14
3	7	11	15
4	8	12	16

Reverse.

Solⁿ ⇒ transpose → reverse each row

Doubt

Q.

① 30 min pen & paper
Brute force Solⁿ

② across the 1st hint

② > 1 hr. 2nd hint

③ Raise the TA request

(examples & observations)

3 2 1 0

13	9	5	1	0
14	10	6	2	1
15	11	7	3	2
16	12	8	4	3

RGB
(0, 255)
(255, 0, 0)
(0, 255, 0)

