

2D Arrays

Hello Everyone

Today's Content:

- 1) Basics of 2D Arrays
- 2) Print row wise sum
- 3) Print column wise sum
- 4) Print Diagonal elements
- 5) Print all element diagonally from right to left.
- 6) Transpose of a matrix.
- 7) Rotate matrix by 90°

Col Index

mat
N*M

0	1	2	
0	(0,0)	(0,1)	(0,2)
1	(1,0)	(1,1)	(1,2)
2	(2,0)	(2,1)	(2,2)

Row Index

mat[2][1]

#row index

Col index

Top right cell \rightarrow mat[0][M-1]

Bottom right cell \rightarrow mat[N-1][M-1]

Q) Given a 2D matrix, print row-wise sum.

The diagram shows a 3x4 matrix with indices 0, 1, 2, 3 for both rows and columns. The elements are numbered 1 through 12. To the right, three arrows point to the results of traversing the matrix in different ways:

- Row 0: 1, 2, 3, 4 → 10 (row-major traversal)
- Row 1: 5, 6, 7, 8 → 26 (row-major traversal)
- Row 2: 9, 10, 11, 12 → 42 (row-major traversal)

Below the matrix, the output (OP) is given as:

OP: 10
26
42

On the right, two formulas are shown:

$$N = \text{Mat.length}$$

$$M = \text{mat[0].length}$$

An orange arrow points from the word "rows" to the length of the first row (M), labeled "no. of columns".

```
for (row=0; row<n; row++)
```

3

Sum = 0

for (c01 = 0; c01 < M; c01++)

1

`SUM = SUM + Mat [row][col];`

println(sum);

TC: $O(N * M)$
SC: $O(1)$

(8) Given a 2D matrix, print ~~row~~^{col}-wise sum.

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

TC: $O(N * M)$
SC: $O(1)$

for (col = 0; col < M; col++)

1

sum = 0

for (row = 0; row < N; row++)

2

sum = sum + Mat[row][col];

printIn(sum);

Op: 15

COL ROW mat[R][c] SUM

0 0

1

1

1

5

$$= 1 + 5 = 6$$

2

9

$$= 6 + 9 = 15$$

1 - - - -

2 - - -

3 - - - -

Q) Given a 2D square matrix, print diagonal.

1. Principal Diagonal : From top left to bottom right.

2. Anti Diagonal : From top right to bottom left.

1	5	8
4	3	1
6	5	2

Principal Diagonal

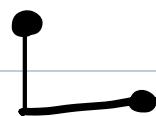
($i = j$)

1	5	8
4	3	1
6	5	2

Anti Diagonal

0,0

1	5	8
4	3	1
6	5	2



Principal Diagonal

no. of rows

```

for (i=0 ; i< n; i++)
2
    print(mat[i][i]);
}

```

OP: 1 3 2

$i=0 \ mat[0][0]$
 $i=1 \ mat[1][1]$
 $i=2 \ mat[2][2]$
 $i=3$

$T.C.: O(n)$
 $SC: O(1)$

	0	1	2	
0	1	5	8	↑ top right corner
1	4	3	1	
2	6	5	2	

Anti Diagonal

$(0, m-1)$

\downarrow
 \leftarrow
 $row++$
 $col--$

$$\begin{array}{r}
 \text{row} \quad \text{col} \\
 8 \rightarrow 0 + 2 = 2 \\
 3 \rightarrow 1 + 1 = 2 \\
 6 \rightarrow 2 + 0 = 2
 \end{array}$$

$$\begin{aligned}
 \text{row} + \text{col} &= N - 1 \\
 \Rightarrow \text{col} &= N - 1 - \text{row}
 \end{aligned}$$

for (row=0; row<n; row++)

}

$$\text{col} = N - 1 - \text{row};$$

 print (mat[row][col]);

TC: O(n)
SC: O(1)

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

Anti Diagonal

O/P: 8 3 6

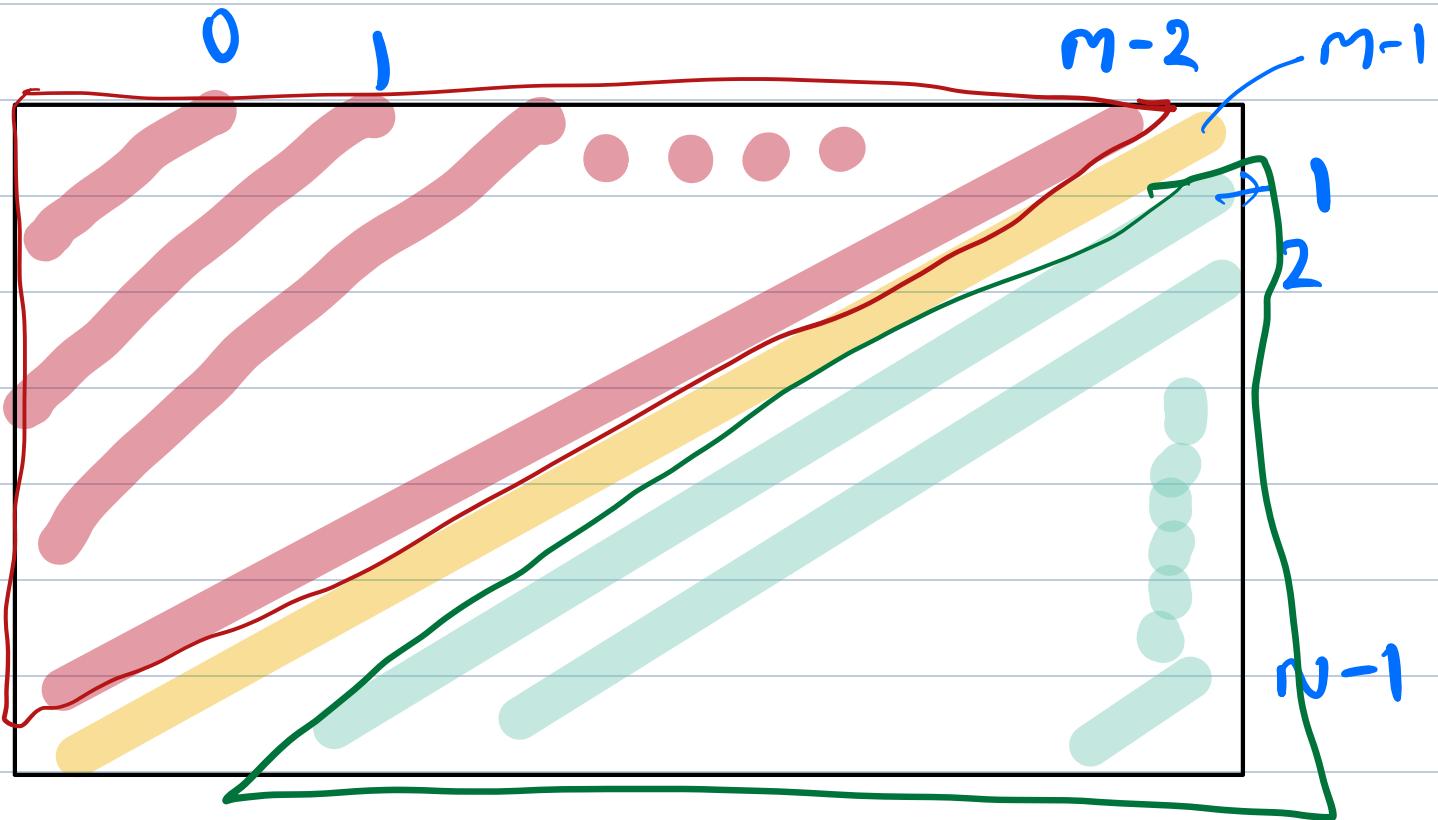
row	col	
0	2	Mat[0][2]
1	1	Mat[1][1]
2	0	Mat[2][0]
3		

Q3 Given a 2D matrix, print all elements diagonally from right to left.

	0	1	2	3
0	5	12	3	1
1	5	6	7	8
2	10	8	2	19

O/P:

5		
12	5	
3	6	10
1	7	8
8	2	



$$[0 \ (M-2)] = M-2 - 0 + 1 = M-1$$

$$[1 \ (N-1)] = N-1 - 1 + 1 = N-1$$

yellow
one

$$\begin{aligned} \text{Total diagonal} &= M-1 + N-1 + 1 \\ &= M+N-1 \end{aligned}$$

* 4 * 5 matrix

$$N = 4 \quad M = 5$$

$$= 475 - 1$$

$$= \underline{\underline{8}}$$

```
for (col = 0; col < m; col++)
{
    row = 0, NC = col
    while (row < n && NC >= 0)
```

$$M = 4$$

```
{     print (mat [row] [NC]);
```

```
    row++;
    NC--
```

```
} }
```

	0	1	2	3
0	5	12	3	1
1	5	6	7	8
2	10	8	2	19

```
for (row = 1; row < N; row++)
{
    NR = row; col = m - 1;
    while (NR < n && col >= 0)
```

```
{     print (mat [NR] [col]),
```

```
    NR++;
    col--
```

```
} }
```

```
} }
```

TC: $O(n * m)$

SC: $O(1)$

Break: 10:43 pm

Q → Find transpose of given **square matrix**

row < col

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

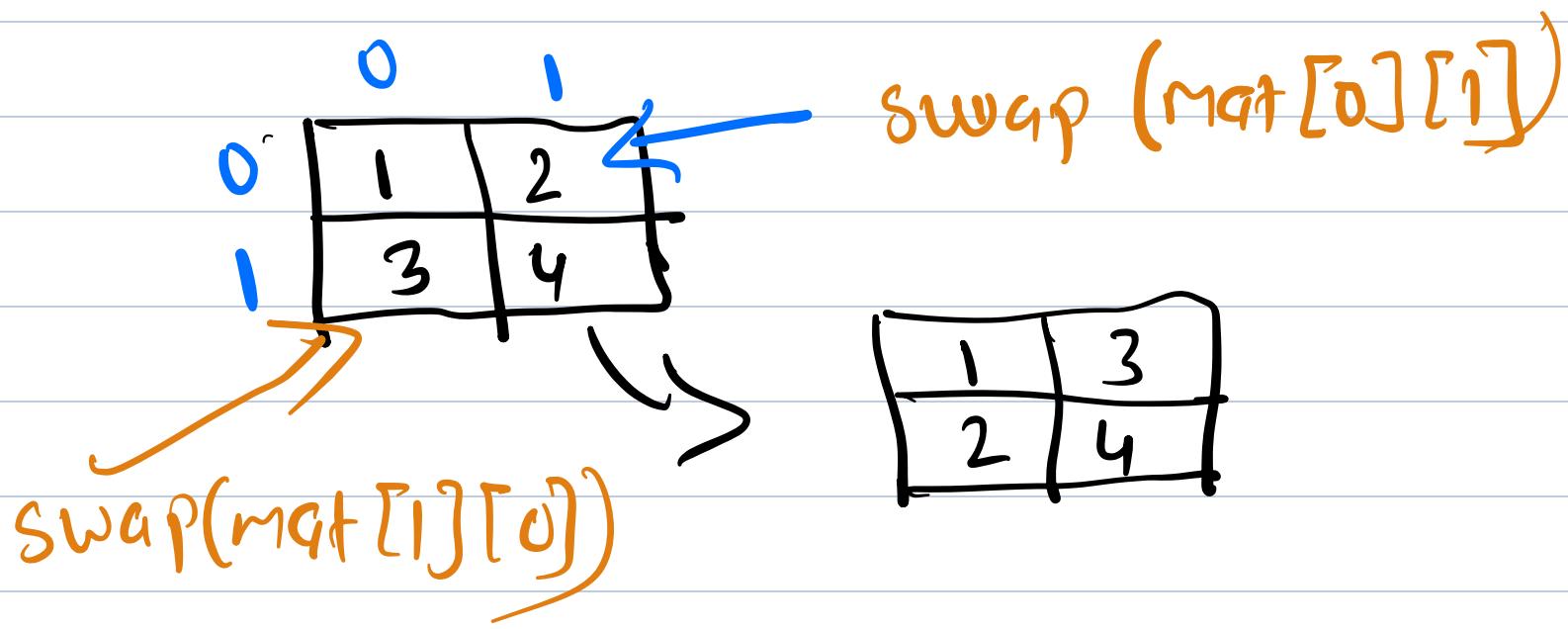
	0	1	2	3
0	1	5	9	13
1	2	6	10	14
2	3	7	11	15
3	4	8	12	16

row > col

1) Row \longleftrightarrow Column ($i, j \leftrightarrow j, i$)

2) Principal diagonal \rightarrow no change.

$\text{swap}(x, y)$
 $\rightarrow \text{temp} = x$
 $x = y$
 $y = \text{temp}$



```

for(row=0; row<n; row++)
{
    for(col=row+1; col<n; col++)
        swap (mat[row][col],
               mat[col][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

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

ROW COL

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

Iteration

ROW COL

0	[1 N-1]	N-1
1	[2 N-1]	N-2
2	[3 N-1]	N-3
3		
:		

$n-1 \dots - - - 1$

$$= 1 + 2 + 3 + \dots + n-3+n-2+n-1$$

$$= \frac{n * (n-1)}{2}$$

TC: $O(n^2)$

SC: $O(1)$

* Rotate the given sq. matrix 90° clockwise.

The diagram illustrates a 4x4 matrix rotation. On the left, a 4x4 grid contains the numbers 1 through 16. An arrow labeled "90°" points from this grid to a second 4x4 grid on the right. The second grid shows the same numbers, but they are arranged such that the original row 0 elements (1, 2, 3, 4) are now in column 0, row 1 elements (5, 6, 7, 8) are in column 1, row 2 elements (9, 10, 11, 12) are in column 2, and row 3 elements (13, 14, 15, 16) are in column 3. This represents a 90° clockwise rotation of the original matrix.

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

1	5	9	-
6	6	5	2
10	8	7	3
13	14	15	16

Hint: What if we first find transpose of matrix.

This diagram shows the steps to rotate a 4x4 matrix 90° clockwise using transpose and reverse rows. It starts with a 4x4 grid where each row is indexed by a blue '0' and each column by a blue '0'. An arrow labeled "Transpose" points to a second 4x4 grid where the columns are swapped (1st column is row 0, 2nd is 1, 3rd is 2, 4th is 3). A blue arrow labeled "↓ reverse row" points to a third 4x4 grid where the rows are reversed (1st row is column 0, 2nd is 1, 3rd is 2, 4th is 3). The final result is a 4x4 grid where the original row 0 elements (1, 2, 3, 4) are now in column 0, row 1 elements (5, 6, 7, 8) are in column 1, row 2 elements (9, 10, 11, 12) are in column 2, and row 3 elements (13, 14, 15, 16) are in column 3.

0	1	2	3
0	1	2	3
1	5	6	7
2	9	10	11
3	13	14	15

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

13	9	5	1
14	10	6	2

The final rotated matrix is shown as a 4x4 grid:

13	9	5	1
14	10	6	2

Output →

15	11	7	3
16	12	8	4

Sol:

- 1) Transpose the matrix } go rotation
- 2) Reverse every row }

Transpose the matrix.

for (row = 0; row < n; row++)

 for (col = row + 1; col < n; col++)

 swap (mat[row][col],
 mat[col][row]);

 }

T.C: $O(n^2)$

for (row = 0; row < N; row++)

{

 for (col = 0; col < N; col++)

 2

 int temp = mat[row][col]

 mat[row][col] = mat[N - 1 - row][N - 1 - col]

 mat[N - 1 - row][N - 1 - col] = temp;

 }

}

TC: O(N²)

row col

0 0 mat[0][0] m[0][3]

0 1 mat[0][1] m[0][2]

1 0 m[1][0] m[1][3]

1 1 m[1][1] m[1][2]

.

.

0	1	2	3
1	5	9	13
2	6	10	14
3	7	11	15

0	1	2	3
1	13	9	5
2	14	10	6
3	15	11	7

0	1	2	3
1	5	1	13
2	6	14	10
3	7	15	11

$$TC = O(N^2 + N^2) = O(N^2)$$

$$SC = O(1)$$

