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[1][1]

mat[2][1]

#row index

#col index

Top right cell → mat[0][M-1]

Bottom right cell → mat[N-1][M-1]

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

	0	1	2	3	
0	1	2	3	4	→ 10
1	5	6	7	8	→ 26
2	9	10	11	12	→ 42

O/P: 10
26
42

no. of rows

N = mat.length
M = mat[0].length

no. of columns

```
for (row=0; row<N; row++)
{
    sum = 0
    for(col = 0; col<M; col++)
    {
        sum = sum + mat[row][col];
    }
    println(sum);
}
```

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

Q) Given a 2D matrix, print 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 ++)
{
    sum = 0
    for(row=0; row<n ; row++)
    {
        sum = sum + Mat [row] [col];
    }
    println(sum);
}
```

O/P: 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

1	5	8
4	3	1
6	5	2

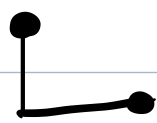
Anti Diagonal

$(i == j)$

0,0

1	5	8
4	3	1
6	5	2

Principal Diagonal



no. of rows

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

o/p: 1 3 2

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

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

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

top right corner

Anti Diagonal

(0, m-1)

row++
col--

	row		col	
8 →	0	+	2	= 2
3 →	1	+	1	= 2
6 →	2	+	0	= 2

$$row + col = N - 1$$

$$\Rightarrow col = N - 1 - row$$

```
for(row=0; row<n; row++)
{
    col= N-1-row;
    print(mat[row][col]);
}
```

TC: O(N)
SC: O(1)

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

Anti Diagonal

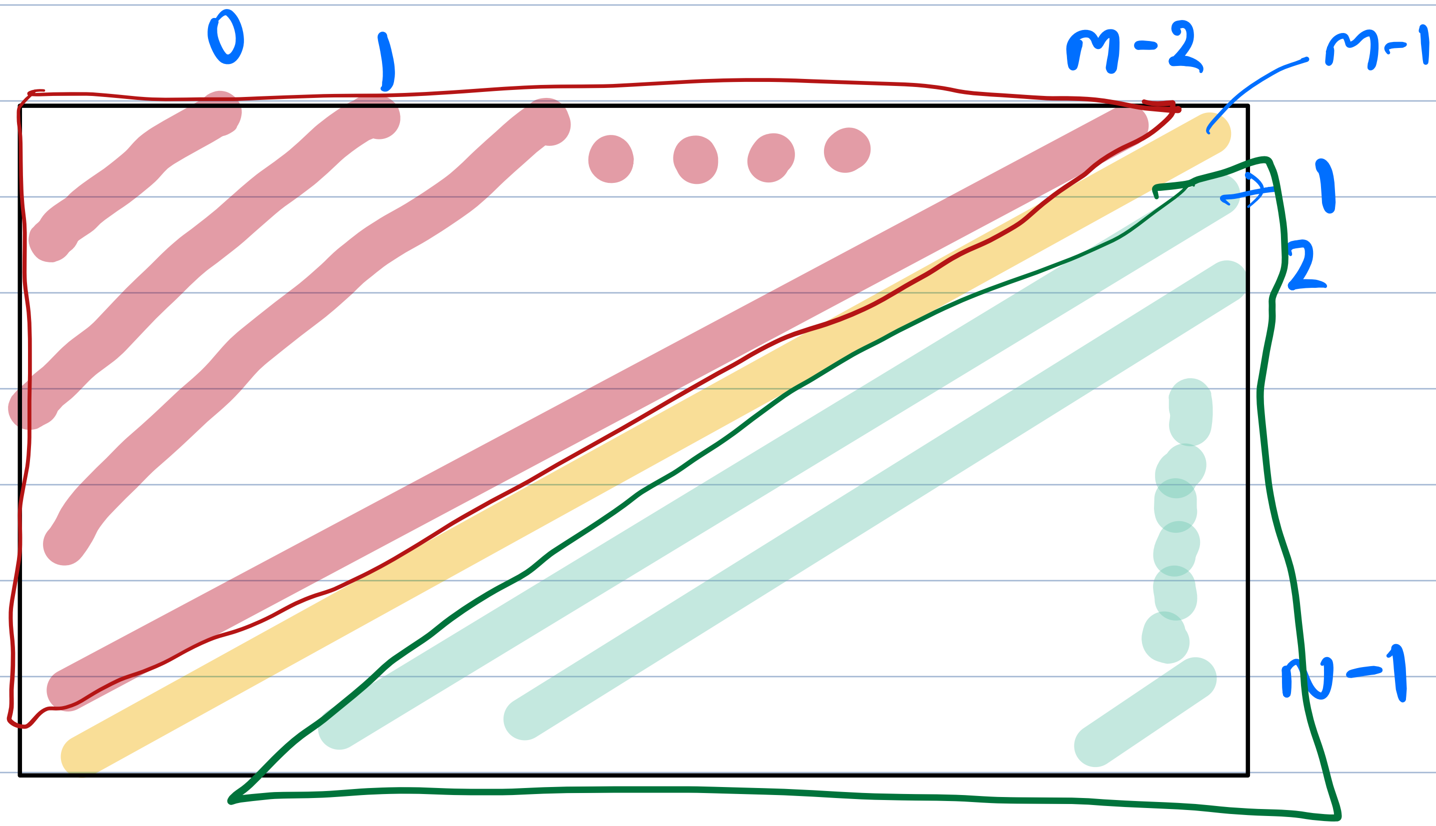
O/P: 8 3 6

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



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

$[1 \quad (N-1)] = N-1-\cancel{1}+\cancel{1} = N-1$

yellow one

Total diagonal $= M-1+N-\cancel{1}+\cancel{1}$

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

* 4 * 5 matrix

$N = 4 \qquad M = 5$

$= 4+5-1$

$= \underline{8}$

```
for(col=0; col<m; col++)
{
   row=0, NC=col
   while(row<n && NC>=0)
   {
       print(mat[row][NC]);
       row++;
       NC--
   }
   println();
}
```

M=4

	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--
    }
    println();
}
```

TC: O(N*M)

SC: O(1)

Break: 10·43 pm

Q→ Find transpose of given square matrix

row<col .

$$
\begin{array}{c|cccc}
 & 0 & 1 & 2 & 3 \\
\hline
0 & 1 & 2 & 3 & 4 \\
1 & 5 & 6 & 7 & 8 \\
2 & 9 & 10 & 11 & 12 \\
3 & 13 & 14 & 15 & 16
\end{array}
\qquad
\begin{array}{c|cccc}
 & 0 & 1 & 2 & 3 \\
\hline
0 & 1 & 5 & 9 & 13 \\
1 & 2 & 6 & 10 & 14 \\
2 & 3 & 7 & 11 & 15 \\
3 & 4 & 8 & 12 & 16
\end{array}
$$

row > col

1) Row ⟷ Column (i,j ⟷ j,i)

2) principal diagonal → no change.

swap(x, y)
→ temp = x
x = y
y = temp

$$
\begin{array}{c|cc} & 0 & 1 \\ \hline 0 & 1 & 2 \\ 1 & 3 & 4 \end{array}
\longrightarrow
\begin{array}{|cc|} 1 & 3 \\ 2 & 4 \end{array}
$$

swap(mat[0][1])

swap(mat[1][0])

n-1

```
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		
⋮		

$N-1 \quad - \; - \; - \; - \quad 1$

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

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

TC: $O(N^2)$

SC: $O(1)$

*) Rotate the given sq. matrix 90° clockwise.

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

$\xrightarrow{90^\circ}$

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

Hint: What if we first find transpose of matrix.

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

$\xrightarrow{\text{Transpose}}$

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

↓ Reverse row

13	9	5	1
14	10	6	2

output →

15	11	7	3
16	12	8	4

Sol:

1) Transpose the matrix
2) Reverse every row

} $90°$ rotation

Transpose the matrix.

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

TC: $O(n^2)$

```
for( row=0 ; row<N; row ++)
{
    for(col=0; col<N/2; col++)
    {
        int temp= mat[row][col]
        mat[row][col] = mat[row][N-1-col]
        mat[row][N-1-col] = temp;
    }
}
```

TC: $O(N^2)$

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

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

$SC = O(1)$