Matrix Multi dimensional array

Hello World

Shrinath
Date
Page No.

~~Imp Spiral~~

Multidimensional array means --> ~~Array of Array~~

~~Day1~~

Aarray में system में row wise feed होती है value.

1st row 2nd row 3rd row

given a matrix m x n

int rows = arr.length; ⇒ length of array
int columns = rows[0].length; ⇒ length of row

H.W. description gfg. question

Day 2

gfg. Sum of upper and lower traingles

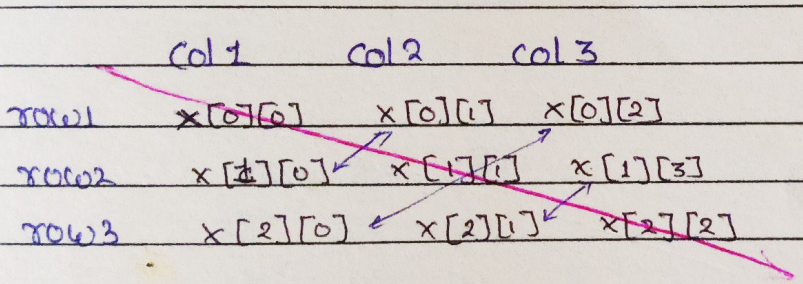
卐 Shrinath
Date
Page No.

Day 3

Transpose of matrix gfg

- transpose of square matrix of size N*N
- N → number of Row and Column
- changing rows to columns and colums to rows

1	1	1	R → C	1	2	3
2	2	2	C → R	1	2	3
3	3	3		1	2	3



swap until diagonal

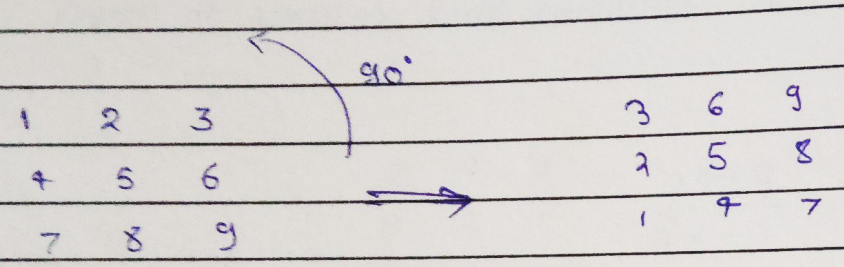
```
for ( i=0; i <= n ; i++ )
for ( j=0; j<i ; j++ )
{ int temp = arr[i][j]
  a[i][j] = a[j][i];
  a[j][i] = temp;
}
```

Date
Page No.

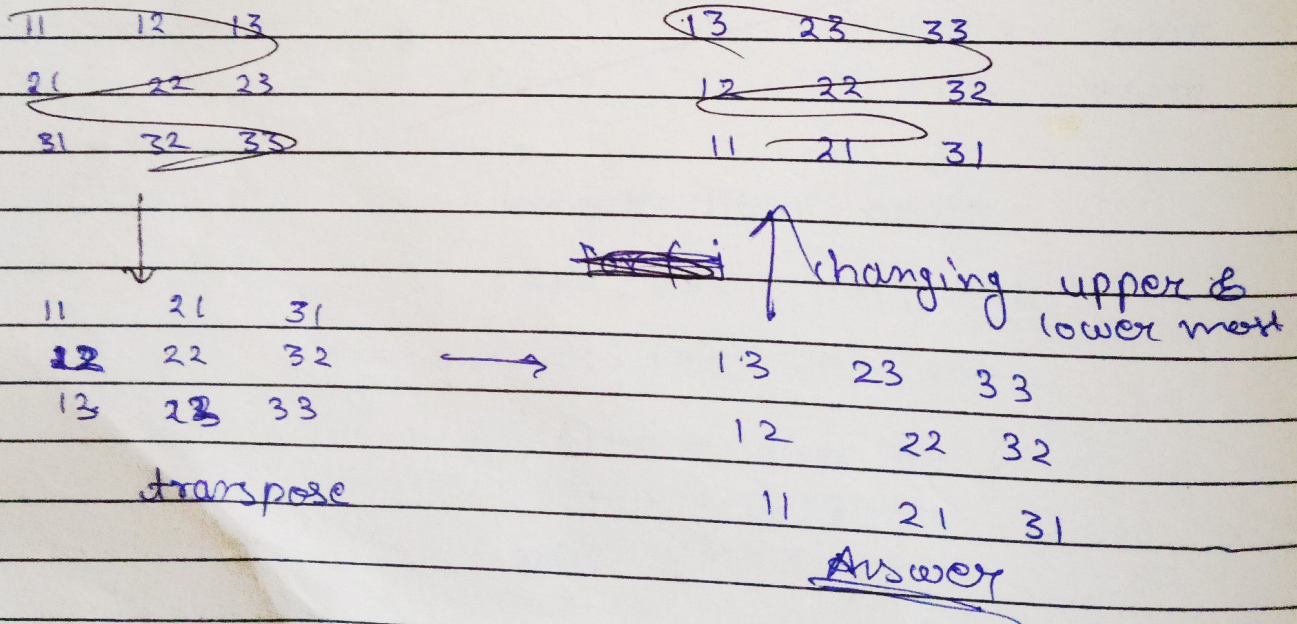
O(n)

Day 7 Rotate by 90° degree + without extra space

Given a square matrix of size N*N.
N → Number of Row and column
Rotate it by 90 degrees in anti-clockwise



input O/P



Shrinath
Date
Page No.

(n)

4×4

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

transpose →

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

↓

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

1	2	3
4	5	6
7	8	9

1	4	7
3	5	8
3	6	9

Shrinath
Date
Page No.

GFG

Q. Given a matrix of R * C, Traverse the matrix in spiral form.

input R = 4 , C = 4

o/p

$$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \end{pmatrix} \Rightarrow$$

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

~~1 2 3 4~~

~~i=0 , j=++ i++ , j~~

11 12 13 14

21 22 23 24

31 32 33 34

41 42 43 44

i j++ , when j<n until

j i++ until i<n

i j-- j>=0

i-- , j i!=j

i , j++

Shrinath
Date
Page No.

left ↓ Right ↓

Top →	1	2	3	4	Top = 0
	5	6	7	8	bottom = R - 1
	9	10	11	12	left = 0
bottom →	13	14	15	16	right = C - 1

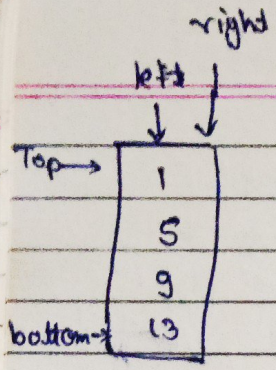
left → right	1	2	3	4	top++ top = 1
top → bottom	8	12	16		right-- right = 2
right → left	15	14	13		bottom-- bottom 2
bottom → top	13	9	5		left++ left 1

while (top <= bottom && left <= right)
continue loop until

{
for (int i = left ; i <= right ; i++) ans.add(matrix[top][i]);
top++ ;

for (i = top ; i <= bottom ; i++) matrix[i][right];
rigght--
if (top <= bottom)
for (i = right ; i >= left ; i--) matrix[bottom][i]
if (left <= right) bottom
for (i = bottom ; i >= Top ; i--) matrix[left][i]
left++ ;

Shrinath
Date
Page No.



Top = 0
bottom = 3
left = 0
right = 0

while (top <= bottom && left <= right)

{

for(left → right) add matrix[Top][i];

Top ++

for (Top → bottom) add matrix[i][right]

right -- ;

top <= bottom 0 <= 3

for

6 12 18
36 10 22
34 28 24

Shrinath
Date
Page No.

Medium

Set matrix zero done 1q.

done 1q. Find first & last position of element in sorted array

1q. Jump game - did not understand 4q.

卐 Shrinath
Date
Page No.

Binary Search in 2D Array

Searching in matrices

linear search target 91

nums

	0	1	2
0	18	9	12
1	36	-9	91
2	99	33	16

```
for(r=0; r<n; r++)
{ for(c=0; c<n; c++)
        if( nums[r][j] = target;
            return new int[] {r, j};
  }
return {-1, -1}
```

Answer [1, 2]

$N * N = N^2$ = time complexity → $O(N^2)$

Q.

The matrix is sorted in a row wise & column wise manner

	0	1	2	3
0	10	20	30	40
1	17	15	35	45
2	28	28	37	49
3	33	34	38	50

target = 37

valid for m×n also case 1 if (nums[r][c] == target) return;

case 2 if (element > target)
        c--;

case 3 if (element < target)
        r++;

Shrinath
Date
Page No.

```
static int[] search(int[][] matrix, int target)
{
    int r = 0;
    int c = matrix.length - 1;

    while( r < matrix.length && c >= 0 )
    {   if ( matrix[r][c] == target )
            return new int[] { r, c };

        if ( matrix[r][c] < target )
            r++;
        else
            c--;
    }
}
```

Q. Search in a Sorted matrix

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

1st approch linear search

2nd approch convert it into a 1 d array and then apply binary search

But we are not going to do this

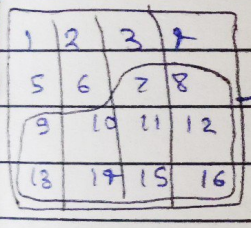
Shrinath
Date
Page No.

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

Tack middle colum & perform binary search on it.

let target 2

mid > target ∵ mid = 6, tar = 2



→ > mid > target

so all these element > target

so we can exclude the rows

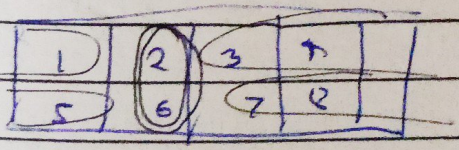
case1 if element = target //ans

case2 if element > target → ignore rows after it

case3 if element < target → ignore above rows

log(N)

in the end 2 rows are remaining.



① cheack whether the mid column you are at contain the answer [2, 6]

② consider the four parts

log(m)

O(log(N) + log(m))