Arrays: 2D Matrices

Java

Syntax

```
int [][] a =    {
        { 10, 20, 30 },
        { 40, 50, 60 }
    };
```

Python

Syntax

```
a =   [
    [10, 20, 30],
    [40, 50, 60]
]
```

	0	1	2
0	10	20	30
1	40	50	60

```
a[1][1]  →  50
a[1][2]  →  60
a[0][1]  →  20
```

```
int [][] a = new int[5][6]
```

Rows Columns

List Comprehension

```
a = [[0] * 6 for i in range(5)]
```

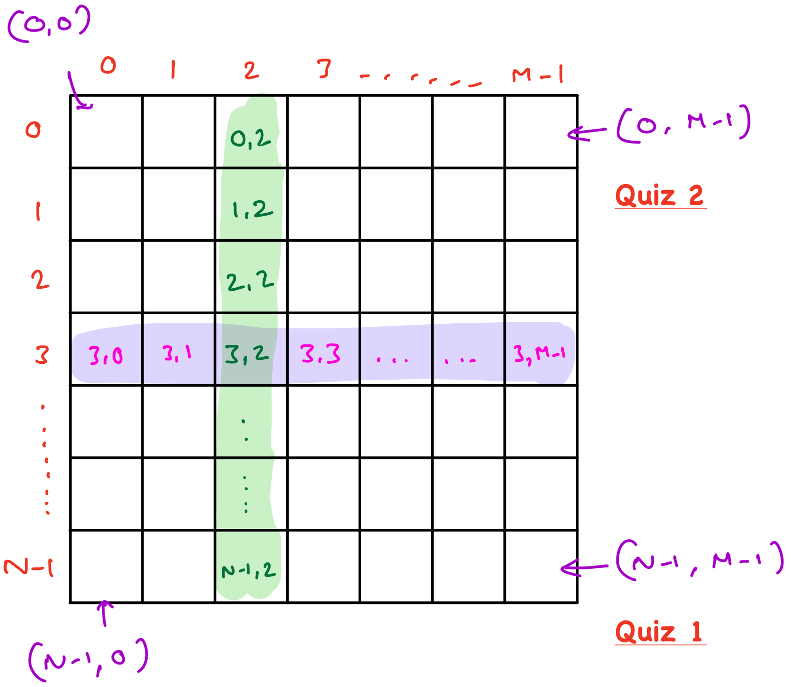
Columns Rows

	0	1	2	3	4	5
0	0	0	0	0	0	0
1	0	0	0	0	. .	.
2	.	.	.	.		
3						
4						

How to declare a matrix of size N * M ?

```
int [][] a = new int[N][M];
                     ↗   ↑
                  Rows   Columns
```

```
a = [ [0] * M   for i in range (N) ]
             ↑                    ↑
          Columns                Rows
```



Quiz 2

Quiz 1

when we are iterating over a column

→ row index goes from (0 to n-1)

when we are iterating over a row

→ col index goes from (0 to M-1)

Quiz 3 Printing a matrix of size N x M

```
for i → [0, N-1] :
    for j → [0, M-1]
        print ( A[i][j])
```

TC : O(N * M)

SC : O(1)

Q1. Given a mat[N][M], print row-wise sum.

mat [3] [4]

$$\begin{array}{c|c|c|c|c|l}
 & 0 & 1 & 2 & 3 & \\
\hline
0 & 3 & 8 & 9 & 2 & \rightarrow 22 \\
\hline
1 & 1 & 2 & 3 & 6 & \rightarrow 12 \\
\hline
2 & 4 & 10 & 11 & 17 & \rightarrow 42 \\
\hline
\end{array}$$

Row - i

Col - j

Iterate over each row & get its sum

```
for (i=0; i<N; i++) {
        // Sum of i^th row

        sum = 0

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

Quiz 4

TC : O(N * M)

SC : O(1)

Q2. Given a mat[N][M], find max column-wise sum.

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

mat [3] [4]

	0	1	2	3
0	3	8	9	2
1	1	2	3	6
2	4	10	11	8
	8	20	23	16

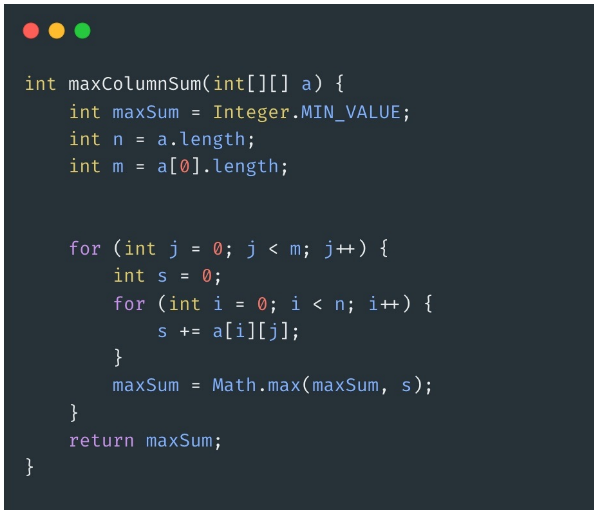
↑ Ans

1) Go col by col & calculate sum
2) Print the max sum

```
ans = -∞
for( j=0; j<M; j++) {
    // sum of j^th col
    sum=0
    for( i=0; i<N; i++) {
        sum += A[i] [j]
    }
    ans = max(ans, sum)
}
print(ans)
```

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

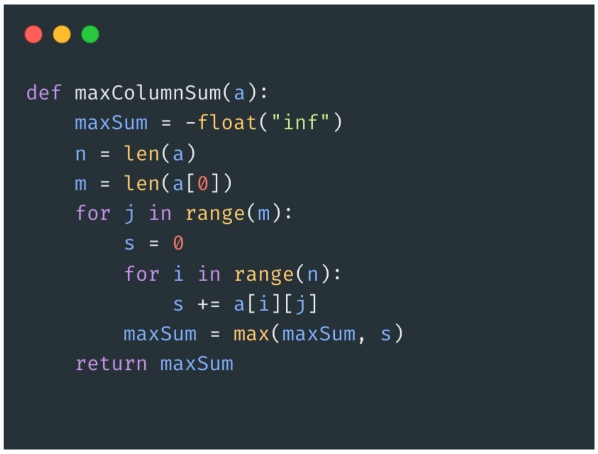
Java



```java
int maxColumnSum(int[][] a) {
    int maxSum = Integer.MIN_VALUE;
    int n = a.length;
    int m = a[0].length;


    for (int j = 0; j < m; j++) {
        int s = 0;
        for (int i = 0; i < n; i++) {
            s += a[i][j];
        }
        maxSum = Math.max(maxSum, s);
    }
    return maxSum;
}
```

Python



```python
def maxColumnSum(a):
    maxSum = -float("inf")
    n = len(a)
    m = len(a[0])
    for j in range(m):
        s = 0
        for i in range(n):
            s += a[i][j]
        maxSum = max(maxSum, s)
    return maxSum
```

Q3 Given a mat[N][N], print diagonal elements.

Square matrix

→ left diagonal

→ Right diagonal

$$\begin{array}{c|cccc} & 0 & 1 & 2 & 3 \\ \hline 0 & 0,0 & & & \\ 1 & & 1,1 & & \\ 2 & & & 2,2 & \\ 3 & & & & 3,3 \end{array}$$

i	j
0	0
1	1
2	2
3	3
4	4

↑ Out of bounds

```
i, j = 0, 0
while ( i < N ) {
        print ( A[i][j] )
        i += 1
        j += 1
}
```

Quiz 5

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

$$\begin{array}{c|cccc} & 0 & 1 & 2 & 3 \\ \hline 0 & & & & 0,3 \\ 1 & & & 1,2 & \\ 2 & & 2,1 & & \\ 3 & 3,0 & & & \end{array}$$

i	j
0	3 ← N-1
+1 ↳ 1	2-1 → 2
+1 ↳ 2	2-1 → 1
3	0

⇓ i<N

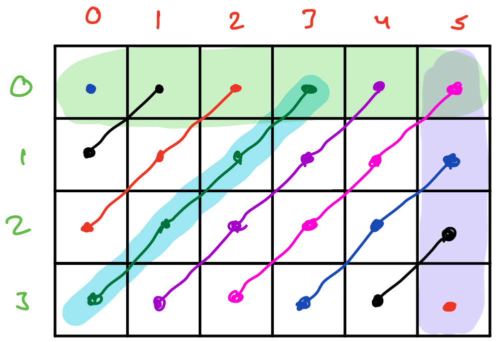
⇓ j>=0

```
i = 0
j = N-1
while ( j >= 0 ) {
      print ( A[i][j] )
      i = i+1
      j = j -1
}
```

Q4 Given a mat[N][M], print diagonal elements going R-L.

Print all R→L diagonals starting from 0th row and (M-1)th col

mat [4] [6]



Row no x	Col no y
0	3
1	2
2	1
3	0
$\boxed{4}$	$\boxed{-1}$

out of bounds

$x < N$ AND $y >= 0$

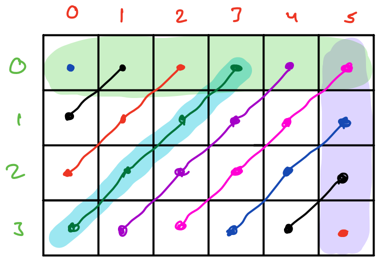
mat [3] [5]

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

Output

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

```
// Iterate over 0th row
for ( j=0 ; j<M ; j++) {
        // Cell [0,j]
        x=0 , y = j
        while ( x<N and y>= 0 ) {
                print (A[x][y])
                x = x+1
                y = y-1
        }
}
```



```
// Iterate over the last col - (M-1)th col
for ( i = 1 ; i<N ; i++ ) {
        // Cell - [i, M-1]
        x = i ,    y = M-1
        while ( x<N and y>= 0 ) {
                print (A[x][y])
                x = x+1
                y = y-1
        }
}
```

To avoid repetition of [0,M-1]

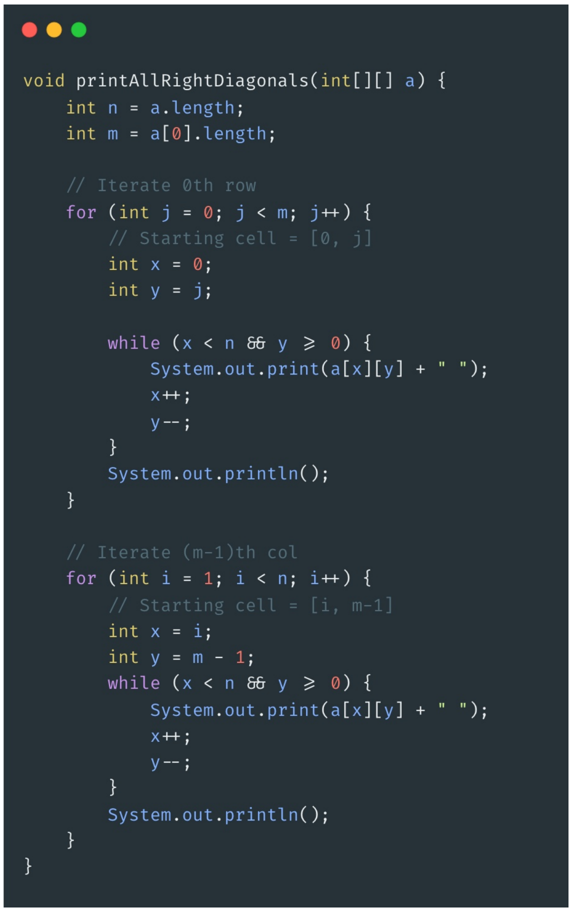
Quiz 6

TC : O(N*M)

SC : O(1)

Break till 10:30 PM

Java



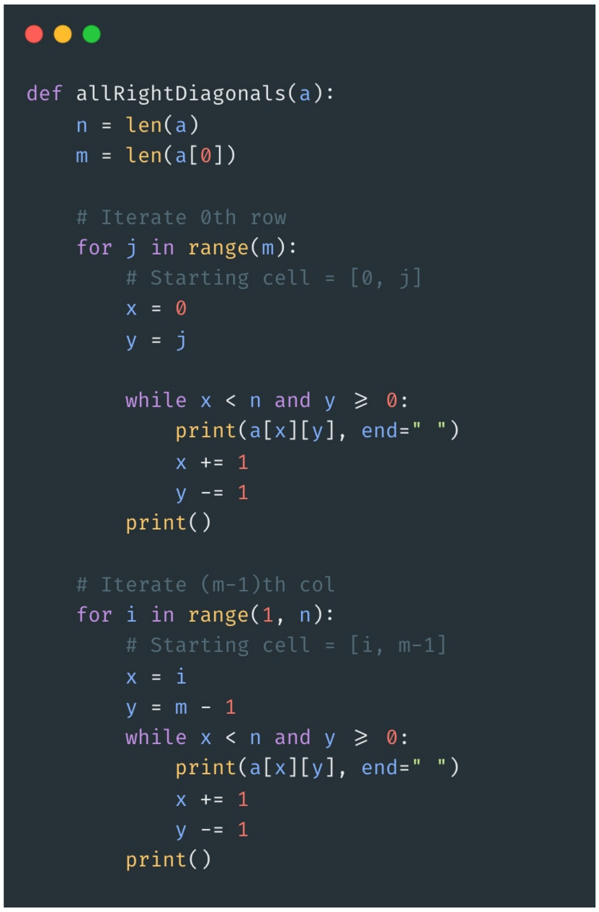
```java
void printAllRightDiagonals(int[][] a) {
    int n = a.length;
    int m = a[0].length;

    // Iterate 0th row
    for (int j = 0; j < m; j++) {
        // Starting cell = [0, j]
        int x = 0;
        int y = j;

        while (x < n && y ≥ 0) {
            System.out.print(a[x][y] + " ");
            x++;
            y--;
        }
        System.out.println();
    }

    // Iterate (m-1)th col
    for (int i = 1; i < n; i++) {
        // Starting cell = [i, m-1]
        int x = i;
        int y = m - 1;
        while (x < n && y ≥ 0) {
            System.out.print(a[x][y] + " ");
            x++;
            y--;
        }
        System.out.println();
    }
}
```

Python



```python
def allRightDiagonals(a):
    n = len(a)
    m = len(a[0])

    # Iterate 0th row
    for j in range(m):
        # Starting cell = [0, j]
        x = 0
        y = j

        while x < n and y ≥ 0:
            print(a[x][y], end=" ")
            x += 1
            y -= 1
        print()

    # Iterate (m-1)th col
    for i in range(1, n):
        # Starting cell = [i, m-1]
        x = i
        y = m - 1
        while x < n and y ≥ 0:
            print(a[x][y], end=" ")
            x += 1
            y -= 1
        print()
```

Q5 Given a mat[N][N], find the transpose inplace.

Square matrix

Change the given array itself

Expected SC: O(1)

mat [5] [5]

Rows ↔ Cols

	0	1	2	3	4
0	1	2	3	4	5
1	6	7	8	9	10
2	11	12	13	14	15
3	16	17	18	19	20
4	21	22	23	24	25

	0	1	2	3	4
0	1	6	11	16	21
1	2	7	12	17	22
2	3	8	13	18	23
3	4	9	14	19	24
4	5	10	15	20	25

mat [0] [1] ⟷ mat [1] [0]

mat [3] [4] ⟷ mat [4] [3]

mat [i] [j] ⟷ mat [j] [i]

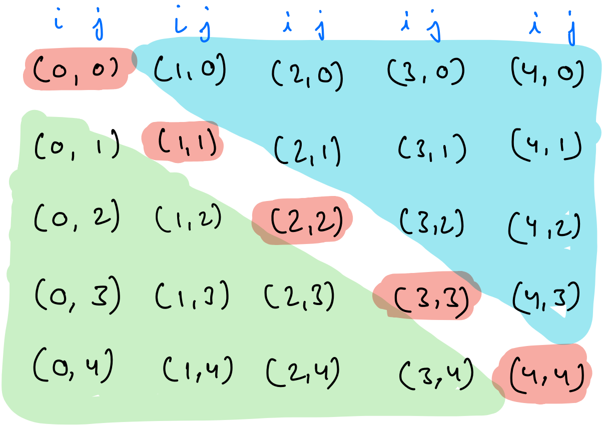
```
for (i=0; i<N; i++) {
    for (j=0; j<M; j++) {
        swap ( mat[i][j], mat[j][i])
    }
}
```

Work ??

Not work

N=5

0,0	↔	0,0
0,1	↔	1,0
0,2	↔	2,0
0,3	↔	3,0
0,4	↔	4,0
1,0	↔	0,1



i j	i j	i j	i j	i j
(0,0)	(1,0)	(2,0)	(3,0)	(4,0)
(0,1)	(1,1)	(2,1)	(3,1)	(4,1)
(0,2)	(1,2)	(2,2)	(3,2)	(4,2)
(0,3)	(1,3)	(2,3)	(3,3)	(4,3)
(0,4)	(1,4)	(2,4)	(3,4)	(4,4)

2 swaps nullify each other

TODO: Write the code on your own

TC : $O(N^2)$

SC: $O(1)$

Quiz 7

Q6 Given a mat[N][N], rotate it by 90 degrees, in clockwise direction.

Expected SC: O(1)

mat [5] [5]

	0	1	2	3	4
0	1	2	3	4	5
1	6	7	8	9	10
2	11	12	13	14	15
3	16	17	18	19	20
4	21	22	23	24	25

Rotate 90° →

4	3	2	1	0	
21	16	11	6	1	0
22	17	12	7	2	1
23	18	13	8	3	2
24	19	14	9	4	3
25	20	15	10	5	4

↓ Transpose

	0	1	2	3	4
0	1	6	11	16	21
1	2	7	12	17	22
2	3	8	13	18	23
3	4	9	14	19	24
4	5	10	15	20	25

Reverse each row →

	0	1	2	3	4
0	21	16	11	6	1
1	22	17	12	7	2
2	23	18	13	8	3
3	24	19	14	9	4
4	25	20	15	10	5

↕ Same

1) Transpose your matrix

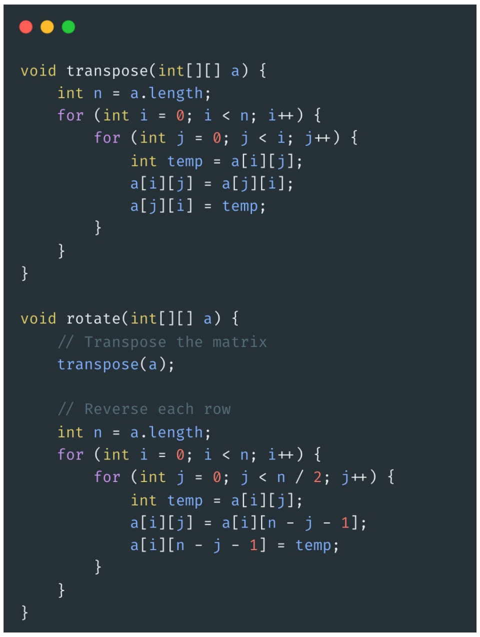
2) Reverse each row

Code - Try on your own

TC : $O(N^2)$

SC : $O(1)$

Java

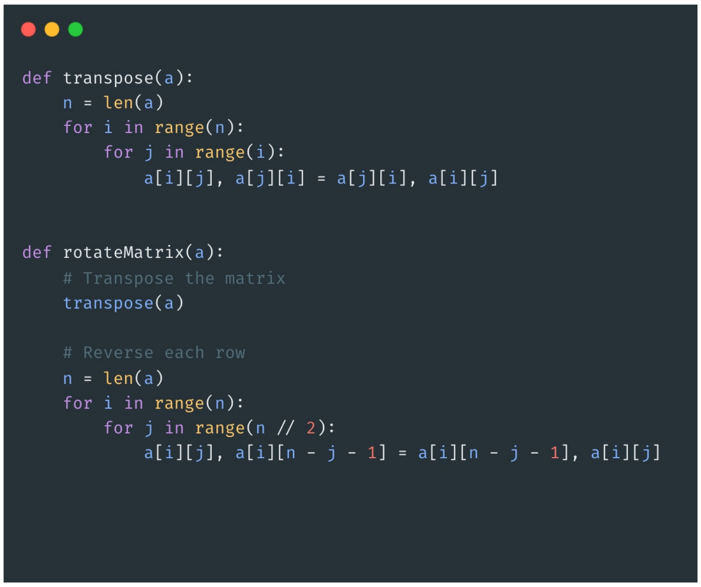


```java
void transpose(int[][] a) {
    int n = a.length;
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < i; j++) {
            int temp = a[i][j];
            a[i][j] = a[j][i];
            a[j][i] = temp;
        }
    }
}

void rotate(int[][] a) {
    // Transpose the matrix
    transpose(a);

    // Reverse each row
    int n = a.length;
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n / 2; j++) {
            int temp = a[i][j];
            a[i][j] = a[i][n - j - 1];
            a[i][n - j - 1] = temp;
        }
    }
}
```

Python



```python
def transpose(a):
    n = len(a)
    for i in range(n):
        for j in range(i):
            a[i][j], a[j][i] = a[j][i], a[i][j]


def rotateMatrix(a):
    # Transpose the matrix
    transpose(a)

    # Reverse each row
    n = len(a)
    for i in range(n):
        for j in range(n // 2):
            a[i][j], a[i][n - j - 1] = a[i][n - j - 1], a[i][j]
```

Doubts

Thank
You

Friday – Contest – 9 PM – 10:30 PM

10:30 PM – Contest discussion class

Coding questions

MCQ s

Extra question

→ Matrix multiplication (Hard)

To attempt a question

→ Understand

→ 5 – 6 examples atleast

→ Brute force

→ Dry run

→ TC & SC analysis

→ If not good enough , then optimize

↳ observations , patterns

↳ Optimised solution

→ Write code

Doubts / Stuck

→ Ask peers

→ Ask TA

→ Access hints / video

→ Doubt session in class

Language syntax doubt

↳ Google

↳ Stackoverflow

↳ ChatGPT

Once a solution is submitted, assume you got 100 points.

Afterwards, accessing hints / videos / solution will not affect your score

If you are stuck on a question for over 30 mins → Ask for help

KPMG Report →

scaler.com / kpmg - report

15 LPA — Base — In hand

+ 10 LPA — Stocks

Good Night

Thank You

Wednesday