

Today's Content,

1. Rotate  $90^\circ$

2. Matrix Multiplication

3. Set zero

18 Transpose of  $\text{mat}[N][N]$  without extra space

Transpose

↳ rows == columns

1<sup>st</sup> row  $\rightarrow$  1<sup>st</sup> col

2<sup>nd</sup> row  $\rightarrow$  2<sup>nd</sup> col

3<sup>rd</sup> row  $\rightarrow$  3<sup>rd</sup> col

;

last row  $\rightarrow$  last col

$\text{mat}[5][5]$

	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

	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

#obs =

$$\text{mat}[i][0] \rightleftharpoons \text{mat}[0][i]$$

$$\text{mat}[3][1] \rightleftharpoons \text{mat}[1][3]$$

$$\text{mat}[4][3] \rightleftharpoons \text{mat}[3][4]$$

# After Transpose  $\text{mat}[i][j] \rightleftharpoons \text{mat}[j][i]$

void transpose(vector<vector<int>> mat) {

int N = mat.size();

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

for(int j = 0; j < N; j++) {

# swap mat[i][j] with mat[j][i]

int temp = mat[i][j];

mat[i][j] = mat[j][i];

mat[j][i] = temp;

}

}

Ex: mat[3][3]

	0	1	2
0	10	20	30
1	40	50	60
2	70	80	90

70

Tracing

i = 0, j = 0: mat[0][0]  $\neq$  mat[0][0]

j = 1: mat[0][1]  $\neq$  mat[1][0]

j = 2: mat[0][2]  $\neq$  mat[2][0]

i = 1, j = 0: mat[1][0]  $\neq$  mat[0][1]

j = 1: mat[1][1]  $\neq$  mat[1][1]

j = 2: mat[1][2]  $\neq$  mat[2][1]

i = 2, j = 0: mat[2][0]  $\neq$  mat[0][2]

j = 1: mat[2][1]  $\neq$  mat[1][2]

j = 2: mat[2][2]  $\neq$  mat[2][2]

# Handle Issues:

Either iterate only on  
upper or lower triangle

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

TC:  $O(N^2)$  SC:  $O(1)$

```
void transpose(vector<vector<int>>& mat) {  
    int N = mat.size();  
    for (int i = 0; i < N; i++) {  
        for (int j = i + 1; j < N; j++) {  
            #swap mat[i][j] with mat[j][i]  
            int temp = mat[i][j];  
            mat[i][j] = mat[j][i];  
            mat[j][i] = temp;  
        }  
    }  
}
```

Q8 Rotate  $\text{mat}[N][N]$  by  $90^\circ$  Clockwise

	0	1	2	3	4			0	1	2	3	4	
0	1	2	3	4	5			0	21	16	11	6	1
1	6	7	8	9	10			1	22	17	12	7	2
2	11	12	13	14	15		Rotate $90^\circ$	2	23	18	13	8	3
3	16	17	18	19	20			3	24	19	14	9	4
4	21	22	23	24	25			4	25	20	15	10	5

# Approach:

1. Calculate Transpose
  2. For every row: Reverse it
- TC:  $O(N^2, N^2)$   
SC:  $O(1)$

↓	0	1	2	3	4			0	1	2	3	4	
0	1	6	11	16	21			0	21	16	11	6	1
1	2	7	12	17	22	<u>Reverse each</u> row →		1	22	17	12	7	2
2	3	8	13	18	23			2	23	18	13	8	3
3	4	9	14	19	24			3	24	19	14	9	4
4	5	10	15	20	25			4	25	20	15	10	5
Transpose								Rotate 90					

vector<vector<int>> Transpose(vector<vector<int>> A){

}

# Matrix Multiplication.

# Rule 1:  $A * B = C$

$$r_1 * c_1 = r_1 * c_2 \quad r_1 * c_2$$

$$\begin{matrix} r_1 & c_1 & & r_2 & c_2 \\ A[3 & 4] & B[4 & 2] & = C_{3*2} \end{matrix}$$

$$A[2 & 5] \quad B[5 & 3] = C_{2*3}$$

$$A[3 & 4] \quad B[5 & 2] = \text{cannot multiply.}$$

$$A(3 \ 4) \quad * \quad B(4 \ 2) \quad = \quad R(3 \ 2)$$
$$\begin{matrix} & 0 & 1 & 2 & 3 \\ 0 & \begin{bmatrix} 1 & 2 & 0 & 1 \end{bmatrix} \\ 1 & \begin{bmatrix} 3 & 2 & 1 & 4 \end{bmatrix} \\ 2 & \begin{bmatrix} -1 & 0 & 1 & 2 \end{bmatrix} \end{matrix} \quad \begin{matrix} B & 0 & 1 \\ 0 & \begin{bmatrix} 2 & 1 \end{bmatrix} \\ 1 & \begin{bmatrix} 1 & 0 \end{bmatrix} \\ 2 & \begin{bmatrix} -1 & 1 \end{bmatrix} \\ 3 & \begin{bmatrix} 2 & -1 \end{bmatrix} \end{matrix} \quad = \quad \begin{matrix} & 0 & 1 \\ 0 & \begin{bmatrix} 6 & 0 \end{bmatrix} \\ 1 & \begin{bmatrix} 15 & 0 \end{bmatrix} \\ 2 & \begin{bmatrix} 1 & -2 \end{bmatrix} \end{matrix}$$

$$R[0,0] = 0^{\text{th}} \text{ Row in } A * 0^{\text{th}} \text{ col in } B = 6$$

$$R[0,1] = 0^{\text{th}} \text{ Row in } A * 1^{\text{th}} \text{ col in } B = 0$$

$$R[1,0] = 1^{\text{th}} \text{ Row in } A * 0^{\text{th}} \text{ col in } B = 15$$

$$R[1,1] = 1^{\text{th}} \text{ Row in } A * 1^{\text{th}} \text{ col in } B = 0$$

$$R[2,0] = 2^{\text{th}} \text{ Row in } A * 0^{\text{th}} \text{ col in } B = 1$$

$$R[2,1] = 2^{\text{th}} \text{ Row in } A * 1^{\text{th}} \text{ col in } B = -2$$

$$R[i,j] = i^{\text{th}} \text{ Row in } A * j^{\text{th}} \text{ col in } B$$

$$A * B = C$$

$$r_1 * c_1 = r_1 * c_2 \quad r_1 * c_2$$

TC:  $O(r_1 * c_2 * c_1)$  or  $O(r_1 c_2 r_2)$  SC:  $O(1)$

$\text{vector} \langle \text{vector} \langle \text{int} \rangle \rangle$  mul ( $\text{vector} \langle \text{vector} \langle \text{int} \rangle \rangle A, \text{vector} \langle \text{vector} \langle \text{int} \rangle \rangle B$ ) {

int  $r_1 = A.size()$ ,  $c_1 = A[0].size()$ ;

int  $r_2 = B.size()$ ,  $c_2 = B[0].size()$ ;

#  $A_{r_1, c_1} * B_{r_2, c_2}$  assume  $c_1 = r_2$  :  $R[r_1][c_2]$

$\text{vector} \langle \text{vector} \langle \text{int} \rangle \rangle R(r_1, \text{vector} \langle \text{int} \rangle (c_2));$  #  $r_1 * c_2$

for (int  $i = 0; i < r_1; i++$ ) {

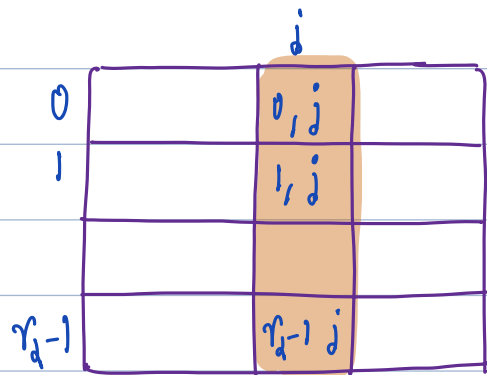
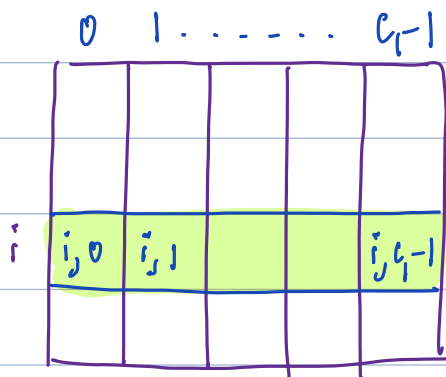
for (int  $j = 0; j < c_2; j++$ ) {

#  $i^{\text{th}}$  Row in  $A * j^{\text{th}}$  col in  $B = \text{res}[i, j]$

/\*

$A[r_1][c_1]$

$B[r_2][c_2]$



long prod = 0;

for (int  $k = 0; k < c_1; k++$ ) {

prod = prod +  $A[i][k] * B[k][j]$

$R[i][j] = \text{prod};$

$k=0: A[i][0] * B[0][j]$

$k=1: A[i][1] * B[1][j]$

$k=2: A[i][2] * B[2][j]$

:

$k=c_1-1: A[i][c_1-1] * B[c_1-1][j]$

return R;