

Today's Content.

1. Rotate 90°
2. Matrix Multiplication
3. Set zero

18 Transport of $\text{mat}(n)(n)$ without intra swap

Transport

↳ rows == columns

1st row \rightarrow 1st col

2nd row \rightarrow 2nd col

3rd row \rightarrow 3rd 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 =

$\text{mat}[1][0] \Leftrightarrow \text{mat}[0][1]$

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

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

After Transport $\text{mat}[i][j] \Leftrightarrow \text{mat}[j][i]$

void transpose(VECTOR & vec, 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 tmp = mat[i][j];

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

 } mat[j][i] = tmp;

}

}

to: mat[3][3]

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

Tracing

i = 0, j = 0 : mat[0][0] \rightarrow mat[0][0]

j = 1 : mat[0][1] \rightarrow mat[1][0]

j = 2 : mat[0][2] \rightarrow mat[2][0]

i = 1, j = 0 : mat[1][0] \rightarrow mat[0][1]

j = 1 : mat[1][1] \rightarrow mat[1][1]

j = 2 : mat[1][2] \rightarrow mat[2][1]

i = 2, j = 0 : mat[2][0] \rightarrow mat[0][2]

j = 1 : mat[2][1] \rightarrow mat[1][2]

j = 2 : mat[2][2] \rightarrow mat[2][2]

#Handle Issues:

Either iterate only in
upper or lower triangle

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

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 tmp = mat[i][j];
        mat[i][j] = mat[j][i];
        mat[j][i] = tmp;
    }
}
```

20. Rotate mat(n)(n) by 90° Clockwise

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

	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

#Approach:

- 1. Calculate transpose
- 2. For every row: Reverse it

$\left. \begin{array}{l} \text{TC: } O(N^2 + N^2) \\ \text{SC: } O(1) \end{array} \right\}$

	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

	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

Transpose

$\xrightarrow{\text{Reverse each row}}$

Rotate 90

vector & vector int > Transporte [vector & vector int > A] {

}

Matrix Multiplication.

Rule 1: $A \times B = C$

$$r_1 \times c_1 = r_2 \times c_2 \quad r_1 + c_2$$

$$\begin{matrix} r_1 & c_1 \\ r_2 & c_2 \end{matrix}$$

$$A[3 \ 4] \quad B[4 \ 2] = C_{3+2}$$

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

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

$$A[3 \ 4] \times B[4 \ 2] = R[3 \ 2]$$

$$\begin{array}{c} A[3 \ 4] \\ \left[\begin{array}{cccc} 0 & 1 & 2 & 3 \\ 1 & 2 & 0 & 1 \\ 3 & 2 & 1 & 4 \\ -1 & 0 & 1 & 2 \end{array} \right] \\ \times \end{array} \begin{array}{c} B[4 \ 2] \\ \left[\begin{array}{cc} 0 & 1 \\ 2 & 1 \\ 1 & 0 \\ -1 & 1 \\ 2 & -1 \end{array} \right] \\ = \end{array} \begin{array}{c} R[3 \ 2] \\ \left[\begin{array}{cc} 0 & 1 \\ 6 & 0 \\ 15 & 0 \\ 1 & -2 \end{array} \right] \end{array}$$

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

$$R[0,1] = 0^{\text{th}} \text{ Row in } A \times 1^{\text{st}} \text{ Col in } B = 0$$

$$R[1,0] = 1^{\text{st}} \text{ Row in } A \times 0^{\text{th}} \text{ Col in } B = 15$$

$$R[1,1] = 1^{\text{st}} \text{ Row in } A \times 1^{\text{st}} \text{ Col in } B = 0$$

$$R[2,0] = 2^{\text{nd}} \text{ Row in } A \times 0^{\text{th}} \text{ Col in } B = 1$$

$$R[2,1] = 2^{\text{nd}} \text{ Row in } A \times 1^{\text{st}} \text{ Col in } B = -2$$

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

$$A \times B = C$$

$$r_1 \times c_1 = r_2 \times c_2 \quad r_1 + c_2$$

$T_{C1} = O(r_1 * c_2 * c_1) \approx O(r_1 c_2 r_2)$ sc: $O(1)$

vector <vector> int \rightarrow mul [vector <vector> int \rightarrow A, vector <vector> int \rightarrow 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 $c_1 = r_2$: $R[r_1][c_2]$

vector <vector> int \rightarrow R(r_1 , vector <vector> int \rightarrow (c_2)); # $\underline{r_1 * c_2}$

for [int $i=0$; $i < r_1$; $i++$] {

 for [int $j=0$; $j < c_2$; $j++$] {

 # ith Row in A + jth Col in B = res[i, j]

 }

$A[r_1][c_1]$

$B[r_2][c_2]$

0 1 ... c_1-1

i

0 1 ... c_1-1

0

1

r_1-1

j

0, j

1, j

r_1-1, j

long prod = 0;

for [int $k=0$; $k < c_1$; $k++$] {

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

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

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

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

:

$R[i][j] = prod;$

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

3

return R;

3