

ROTATE MATRIX BY 90°

★ In this question, we are supposed to rotate a square matrix by 90° clockwise.

So

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

By observing

$a[i][j] \rightarrow a[j][N-i-1]$	$a[i][j] \rightarrow a[j][N-i-1]$
$a[0][0] \rightarrow a[0][3]$	$a[1][0] \rightarrow a[0][2]$
$a[0][1] \rightarrow a[1][3]$	$a[1][1] \rightarrow a[1][2]$
$a[0][2] \rightarrow a[2][3]$	$a[1][2] \rightarrow a[2][2]$
$a[0][3] \rightarrow a[3][3]$	$a[1][3] \rightarrow a[3][2]$

So in our brute force approach, we swap i, j elements with j & $n-i-1$ where n is order of the matrix. Here an extra space is being used for an answer matrix.

Optimal solution is to transpose the matrix and reverse every row.

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

Time Complexity is $O(N^2)$ for transpose
and $O(4 \times \frac{N}{2}) = O(2N)$ for reversing
each row

Pseudocode :

```
rotateMatrix(arr, N) {  
    for(i = 0 → N-1) {  
        for(j = i+1 → N) {  
            swap(arr[i][j], arr[j][i])  
        }  
    }  
    for(i = 0 → N) {  
        for(j = 0 → N/2) {  
            swap(arr[i][j], arr[i][N-j-1])  
        }  
    }  
    return arr  
}
```