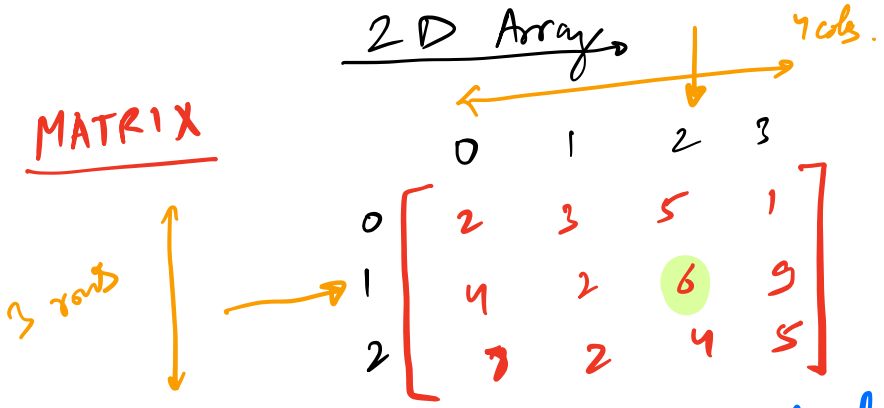
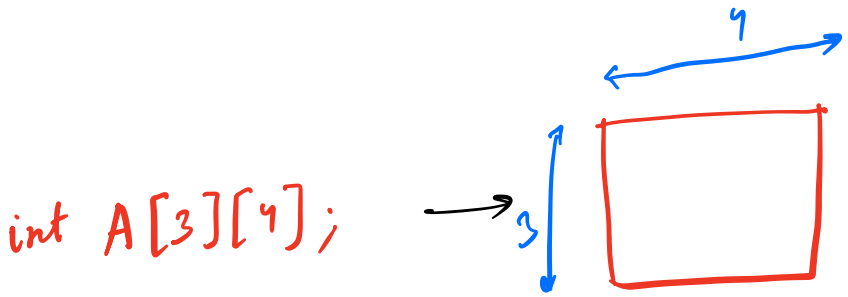


id : (row-id, col-id)



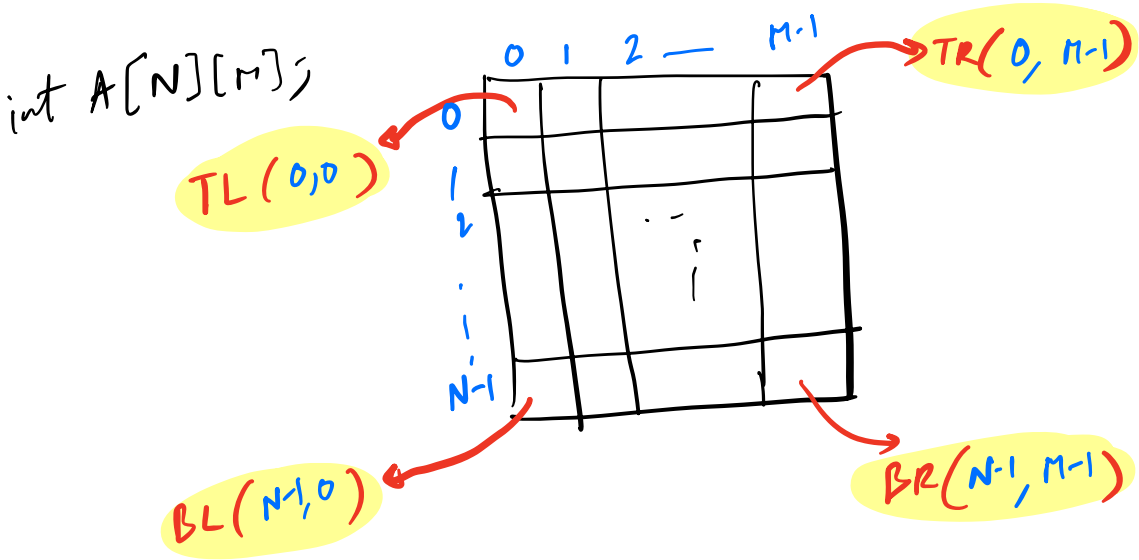
① print the element in i^{th} row, j^{th} col!

print(A[i][j]) → O(1)

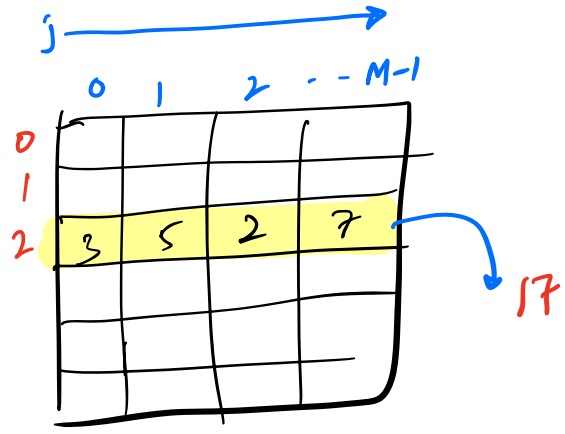
② Assign x to i^{th} row, j^{th} col!

A[i][j] = x → O(1)

int A[N][M];



Q Given a 2D array!
Find the sum of elements of the 2nd row!



2nd row

(2,0), (2,1) (2,2) ... (2,M-1)

| + | + | + - - + |

sum = 0

```
for(j=0; j<M; j++){
    sum += A[2][j];
}
rt sum;
```

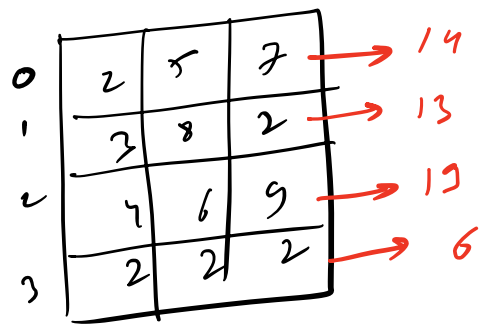
CONVENTION
i: row idx
j: cl idx

TC: O(M)

SC = O(1)

Q. Given a 2D array.
Find the sum of every row!

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



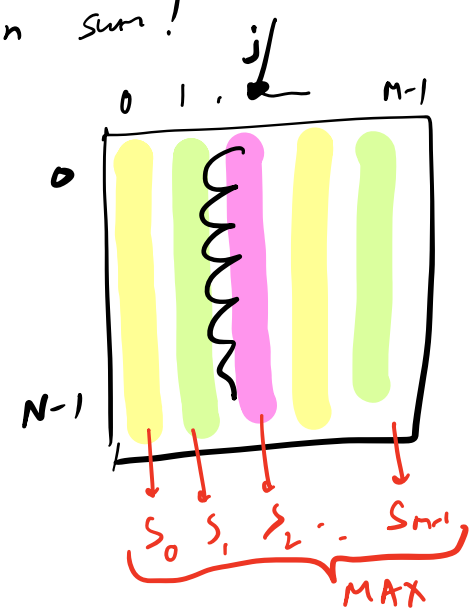
TC = O(NM)

SC = O(1)

Q. Given a matrix A[N][M].
Find the MAX column sum!

ANS = -∞

```
for(j=0; j<M; j++){
    sum=0;
    for(i=0; i<N; i++){
        sum += A[i][j];
    }
    ANS = MAX(ANS, sum);
}
ret ANS;
```



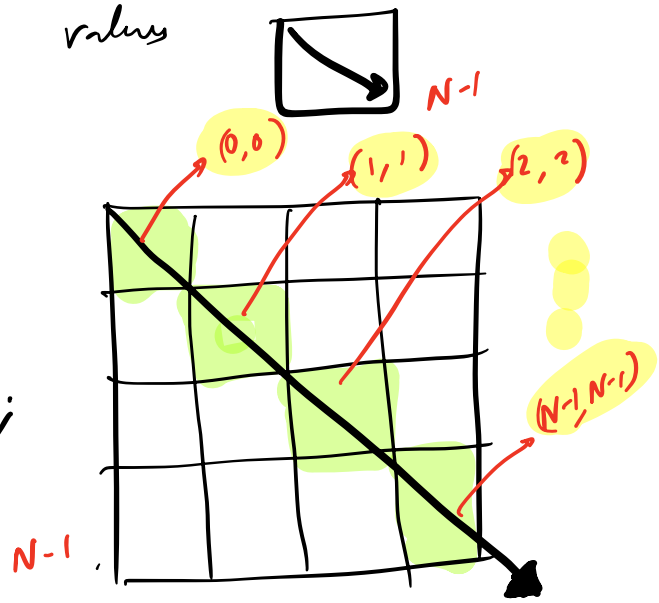
TC = O(NM)

SC = O(1)

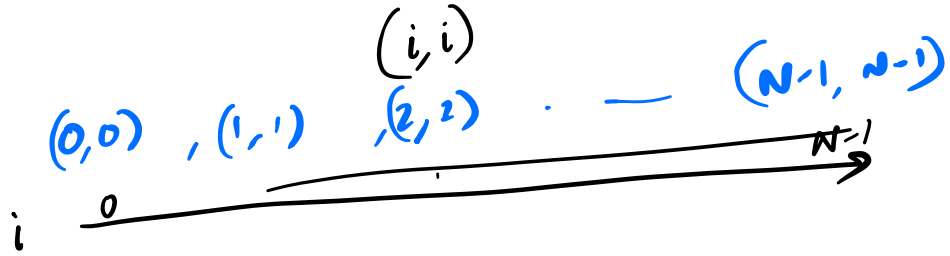
Q Given a 2D array of size N x N print the diagonal values

(i == j)

```
for(i = 0 ⟶ N-1){
    for(j = 0 ⟶ N-1){
        if(i == j){
            print(A[i][j]);
        }
    }
}
```



TC = O(N²)



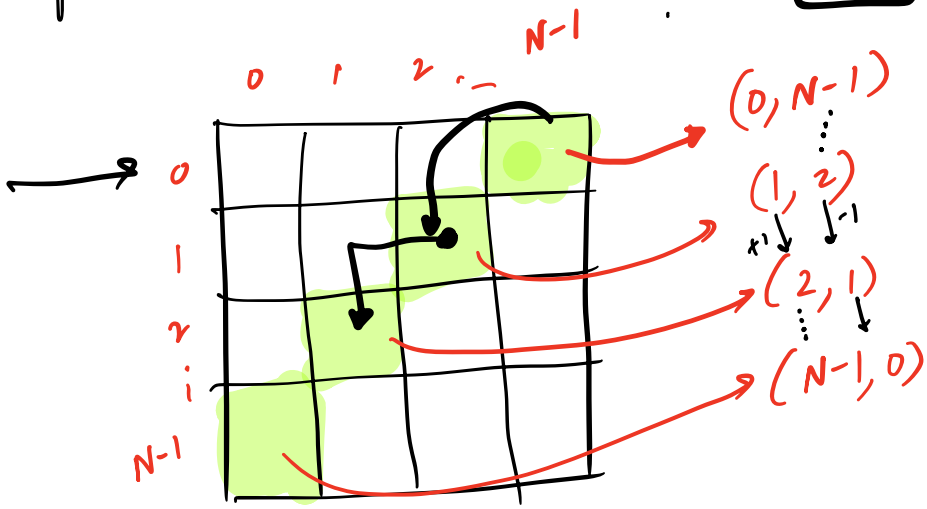

```
for(i = 0 ⟶ N-1){
    print(A[i][i]);
}
```

TC = O(N)

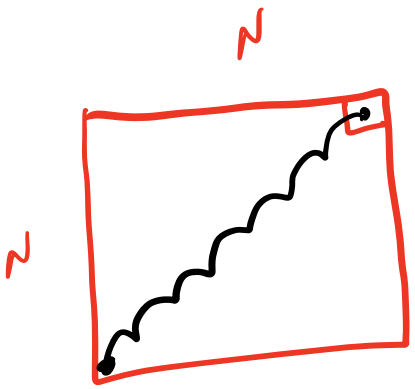
SC = O(1)

1 Given a 2D array of size N x N

print the diagonal values




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

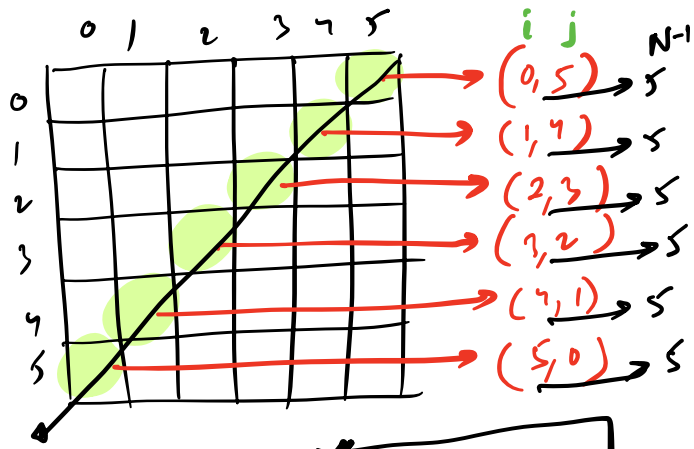


TC = O(N)

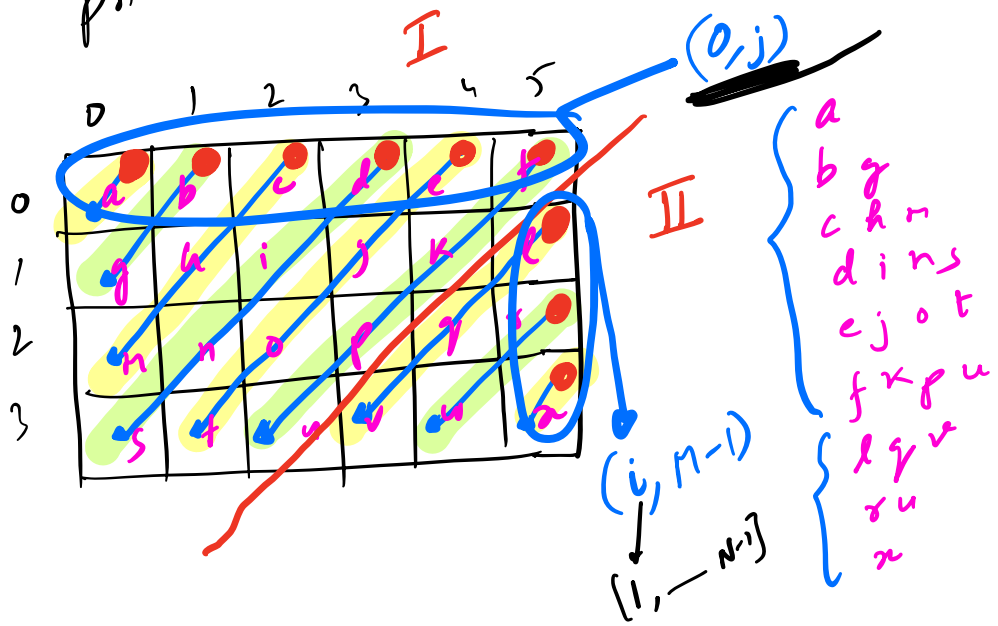
SC = O(1)

N=6

```
for(i=0 → N-1){
    j = N-1-i;
    print(A[i][j]);
}
```



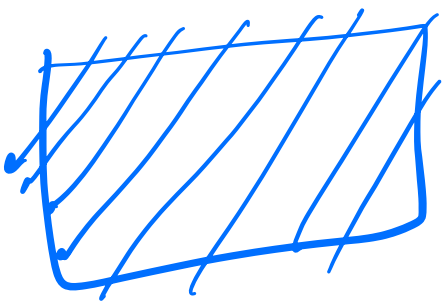
Q Given a 2D array A[N][M] print ALL the diagonals (R→L, T→B)



```
for (j = 0; j < M; j++) {
    I = 0, J = j;
    while (I < N && J >= 0) {
        print( A[I][J] )
        I++;
        J--;
    }
}
for (i = 1 ⟶ N-1) {
    I = i, J = M-1;
    while (I < N && J >= 0) {
        print( A[I][J] )
        I++;
        J--;
    }
}
}
```

I

II

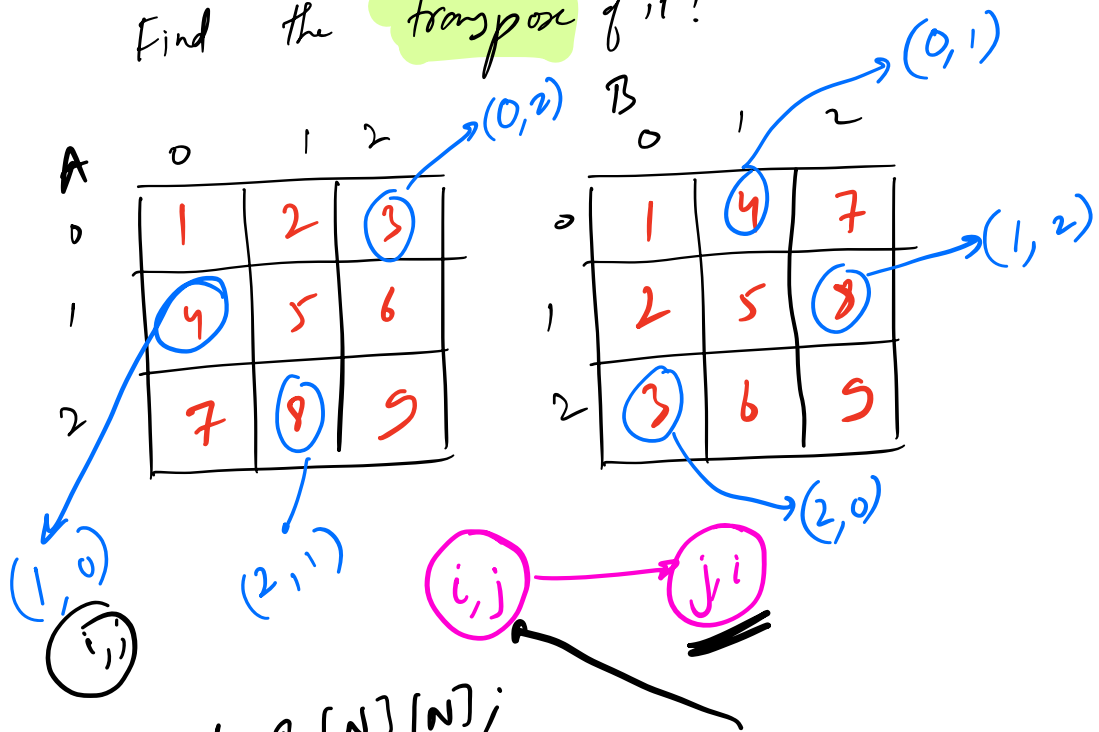


TC = O(NM)

SC = O(1)

Q Given a square matrix A → N×N

Find the transpose of it!



I

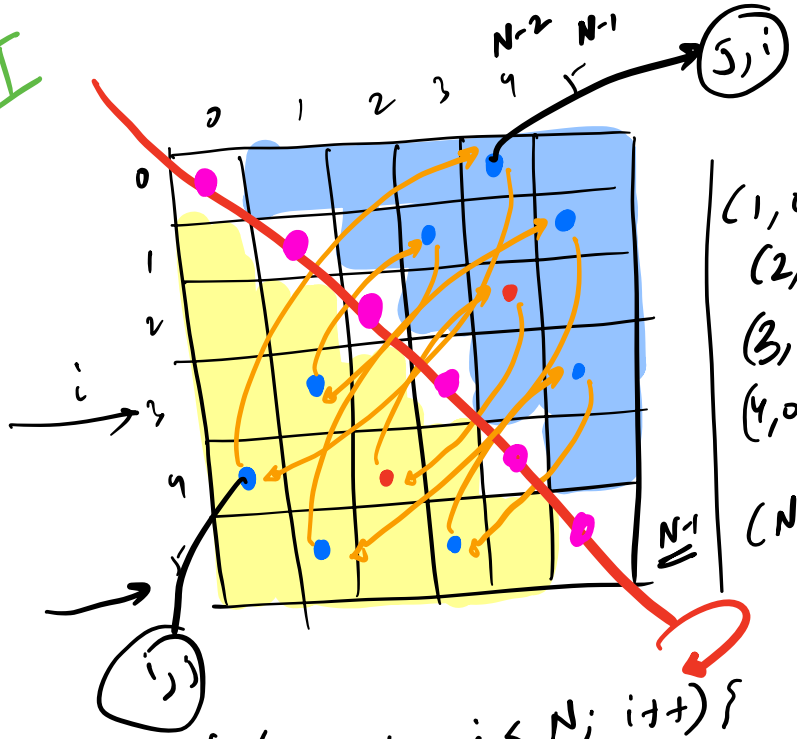
```
int B[N][N];
for( i = 0 ⟶ N-1) {
    for( j = 0 ⟶ N-1) {
        B[j][i] = A[i][j];
    }
}
Copy B[][] ⟶ A[][]
out A[][]
```

TC = O(N²)

S C = O(N²)

O(1)

II




```
for( i = 1; i < N; i++) {
    for(j = 0; j < i; j++) {
        swap(A[i][j], A[j][i]);
    }
}
```

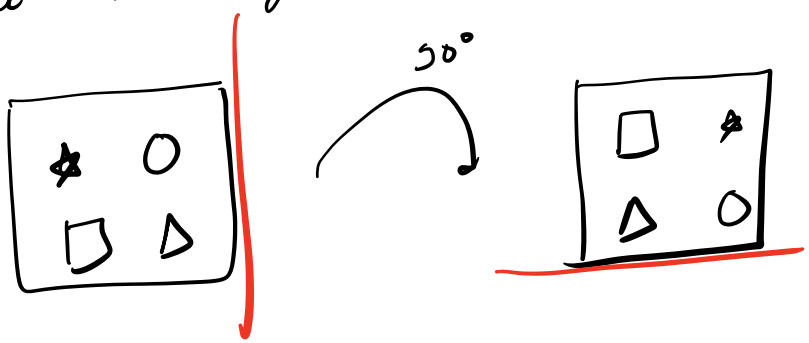
TC = O(N²)

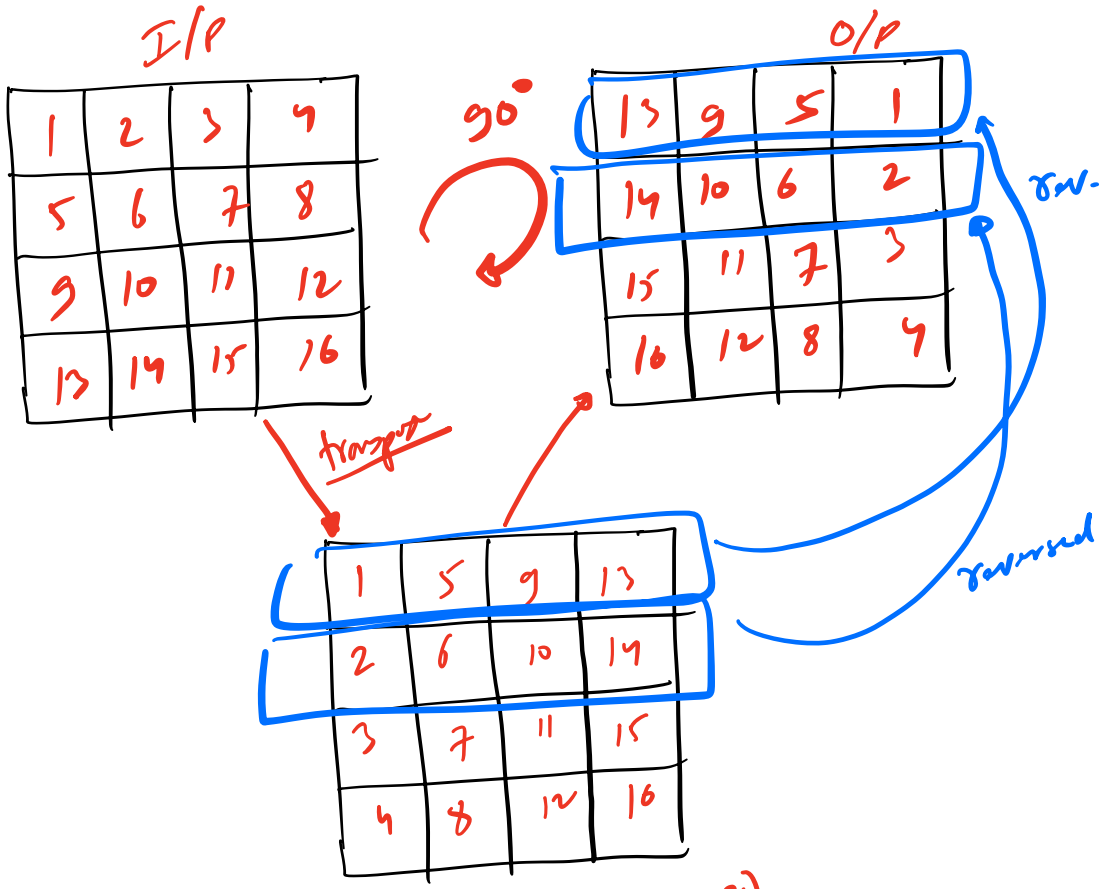
SC = O(1)

INPLACE

I Given a square matrix → N×N

Rotate it by 90° clockwise !



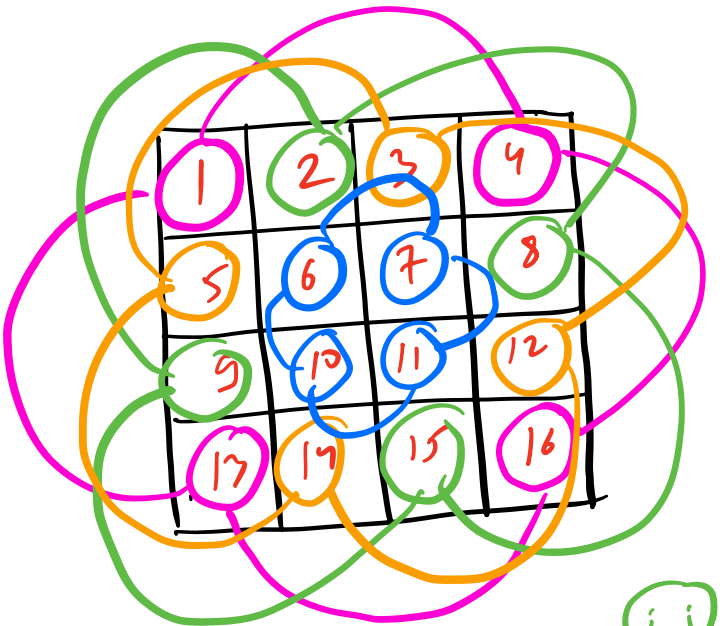


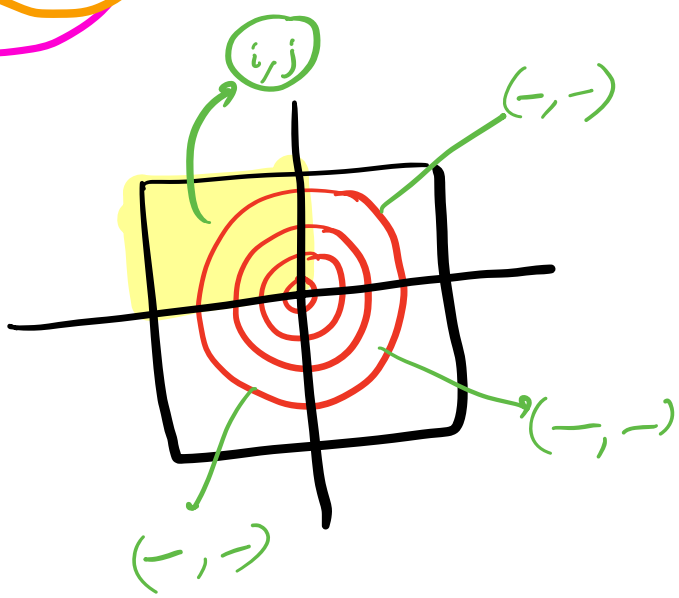
STEPS:

1. find transpose ! → $O(N^2)$
2. Reverse every row! → N^2

$TC = O(N^2)$

$SC = O(1)$


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


i,j
(-,-)
(-,-)
(-,-)