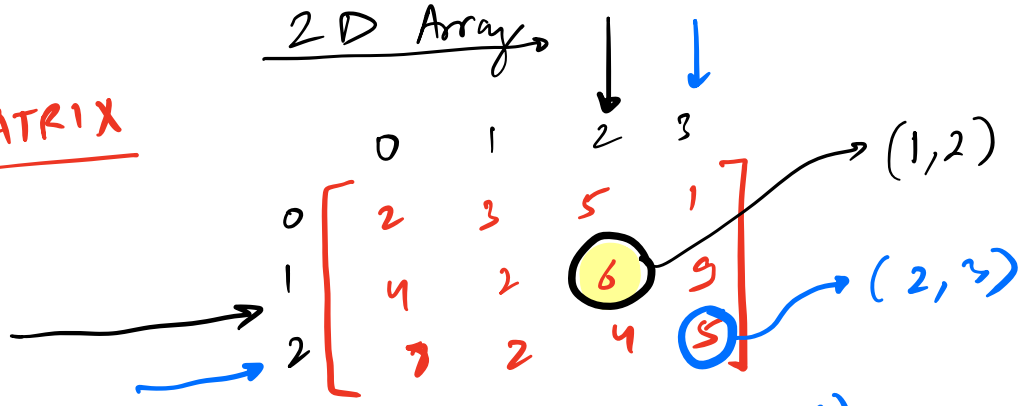
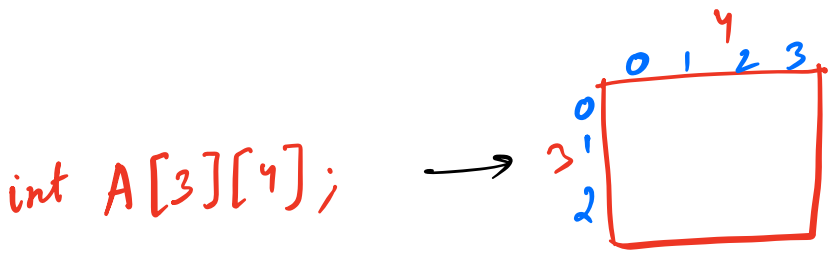
2D Array

MATRIX



id : (row_id, col_id)



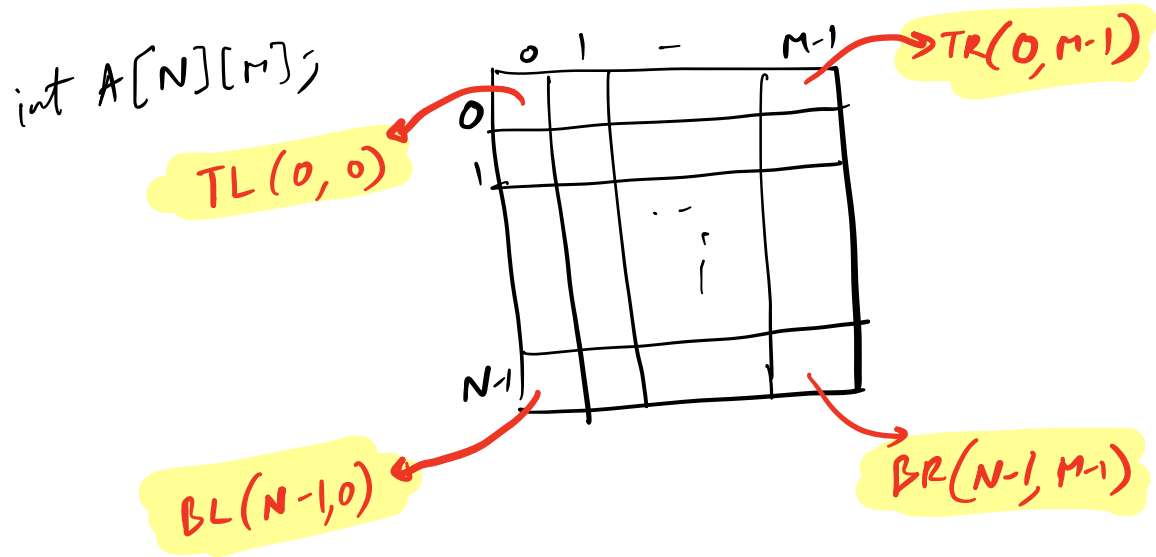
① print the element in ith row, jth col!

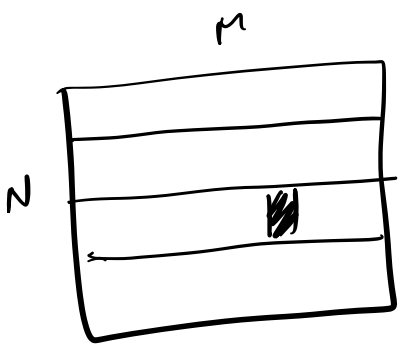
```
print( A[i][j] );
```

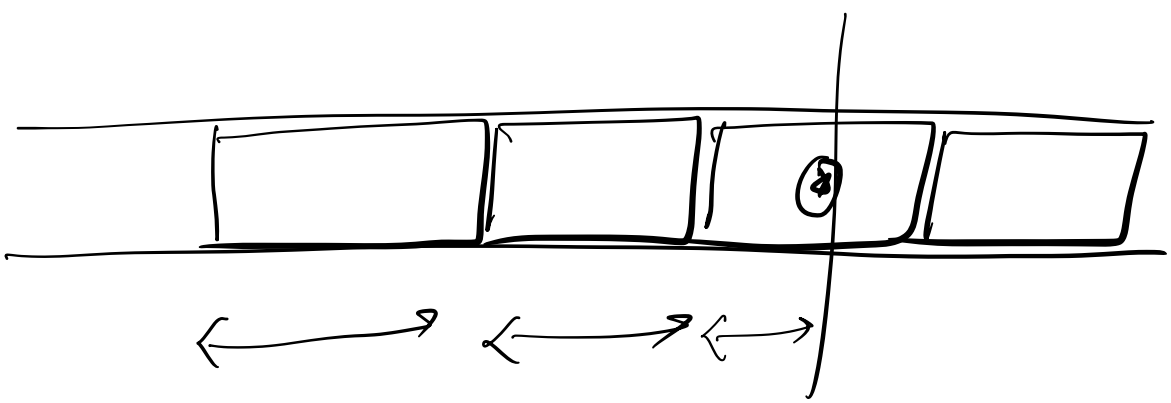
O(1) Access time!

② Assign x to ith row, jth col!

```
A[i][j] = x
```


int A[N][M];
0 1 – M-1
0
1
N-1
TL(0,0)
TR(0,M-1)
BL(N-1,0)
BR(N-1,M-1)


M
N

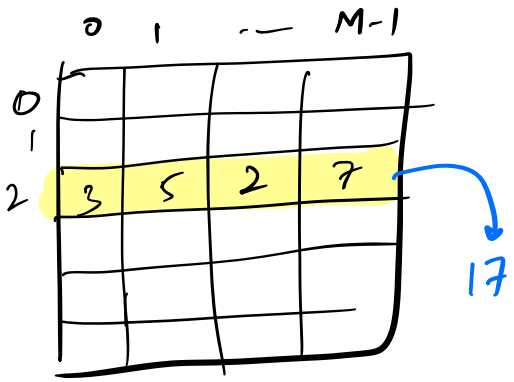


Q Given a 2D array!
Find the sum of elements of the 2nd row!

2nd row

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

0 ——— ++ ——— M-1




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

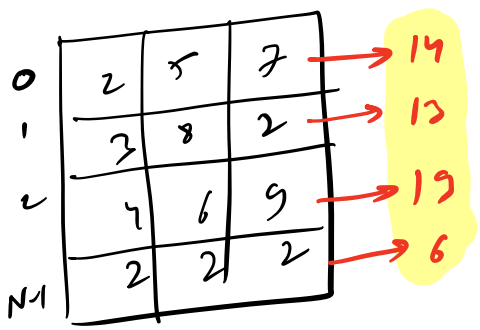
CONVENTION
i: ROW NO
j: COL NO.

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)
}
```

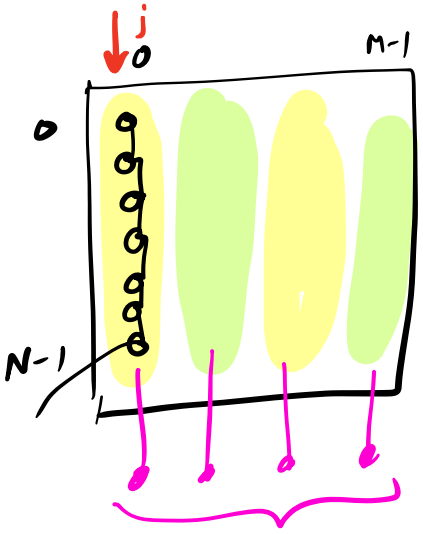


TC = O(NM)

SC = O(1)

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

```
mS = -∞;
for(j=0; j<M; j++){
    sum=0;
    for(i=0; i<N; i++){
        sum += A[i][j];
    }
    if(sum > mS){
        mS = sum;
    }
}
→ ret mS;
```

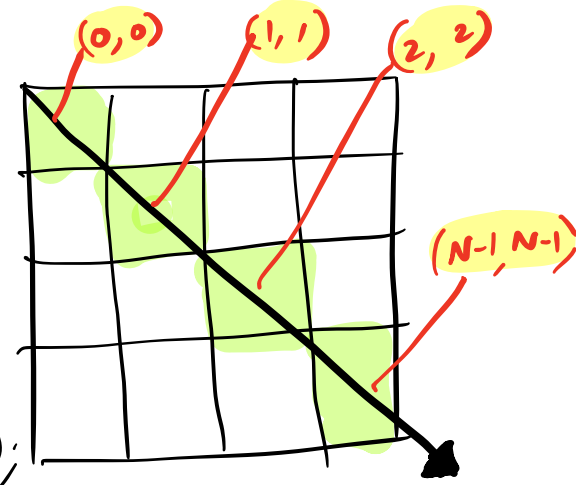


TC = O(NM)

SC = O(1)

Q Given a 2D array of size N × 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)

(0,0) , (1,1) , (2,2) . — (N-1, N-1)

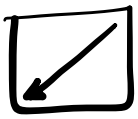
i : 0 ⟶ N-1 ⟶ (i, i)

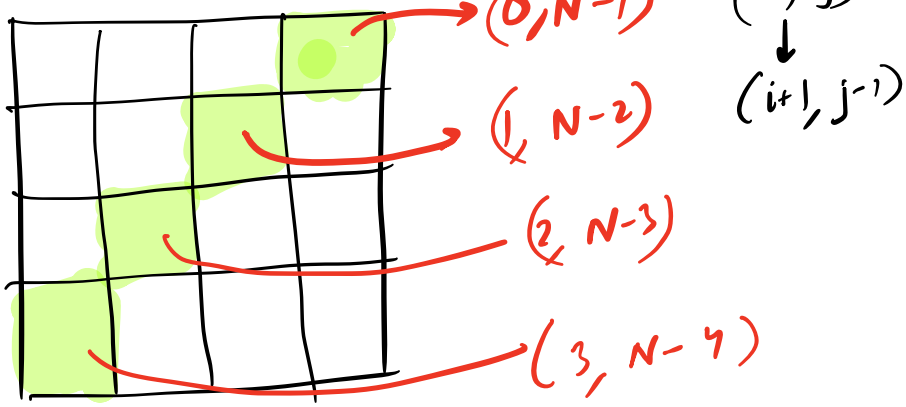
```
for( i = 0; i < N; i++) {
    print( A[i][i]);
}
```

TC = O(N)

SC = O(1)

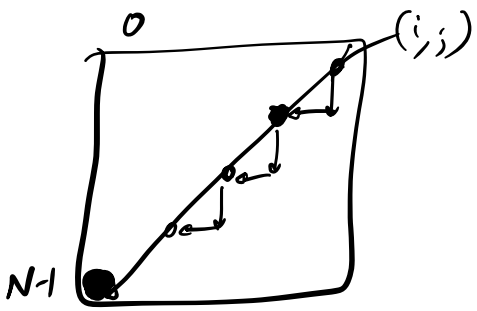
Q Given a 2D array of size N×N print the diagonal values





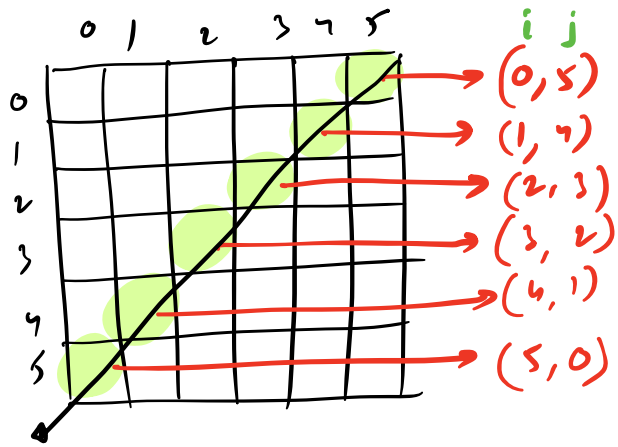
Any one cond. is fine!

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



TC = O(N)

SC = O(1)



$i : 0 \longrightarrow N-1$

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

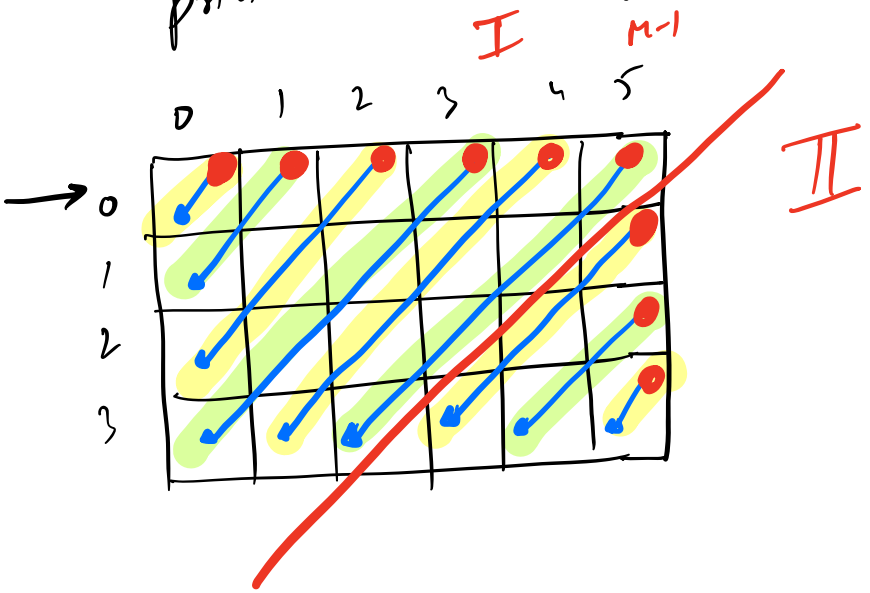
$i+j = N-1$

$j = N-1-i$

$TC = O(N)$

$SC = O(1)$

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

I

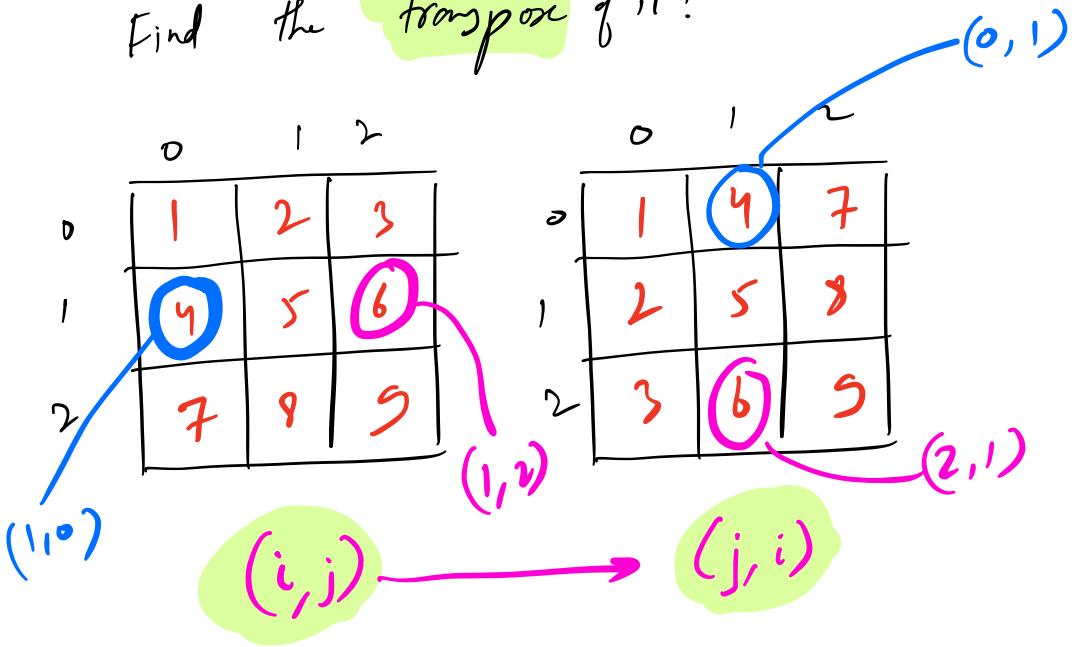
II

TC = O(NM)

SC = O(1)

N

Q Given a square matrix A → N×N
Find the transpose of it!



I

```
int B[N][N];
for(i=0; i<N; i++){
    for(j=0; j<N; j++){
        B[i][j] = A[j][i];
    }
}
copy B → A
```

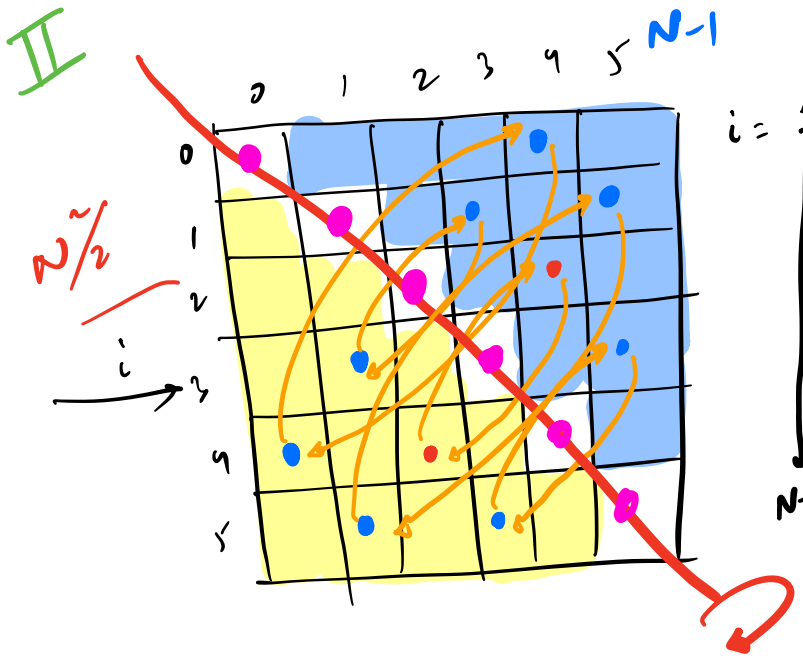
N^2

N^2

TC = $O(N^2)$

SC = $O(N^2)$

II




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

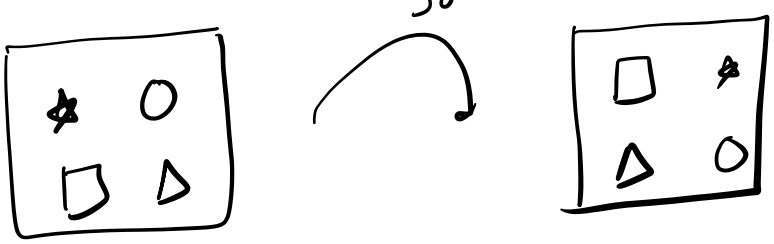
TC = O(N²)

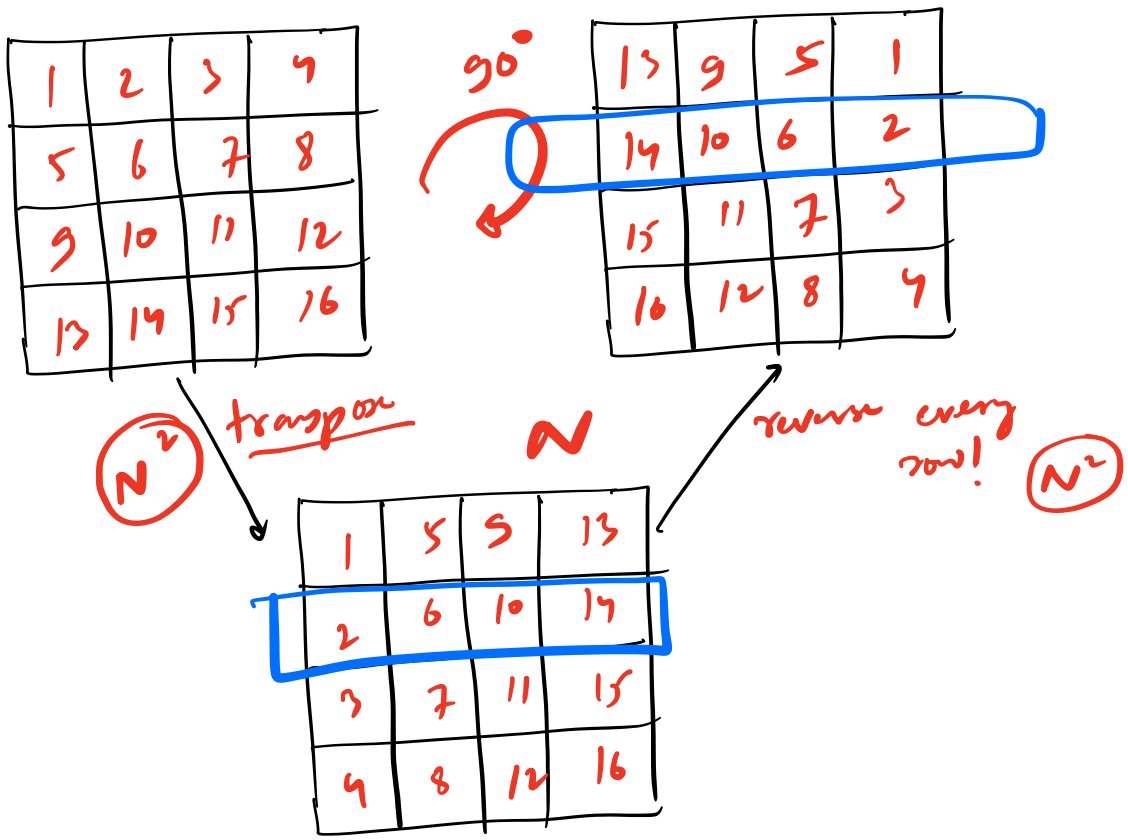
SC = O(1)

IN PLACE

Q Given a square matrix → N×N

Rotate it by 90° clockwise !





$TC = O(N^2)$

$SC = O(1)$

INPLACE

I