867. Transpose Matrix

Easy 3070 427 Add to List Share

Given a 2D integer array `matrix`, return *the* ***transpose*** *of* `matrix`.

The **transpose** of a matrix is the matrix flipped over its main diagonal, switching the matrix's row and column indices.

2	4	-1
-10	5	11
18	-7	6

2	-10	18
4	5	-7
-1	11	6

Row <=> Col.

→ 1 2 3

→ 4 5 6

→ 7 8 9

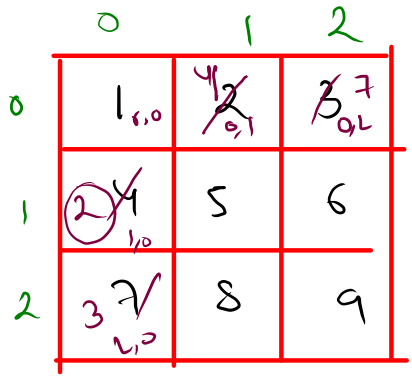
1 4 7

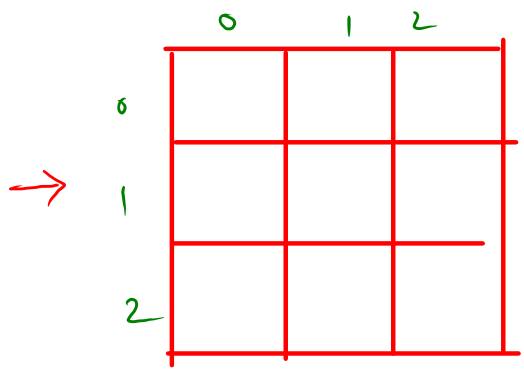
2 5 8

3 6 9

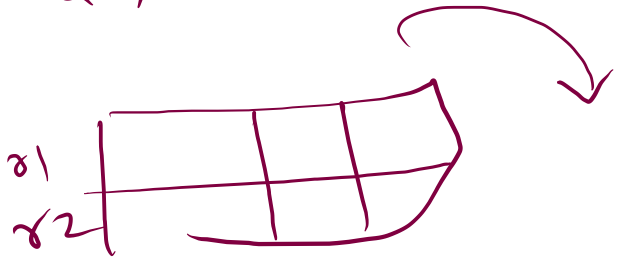
i<j 0 0<1

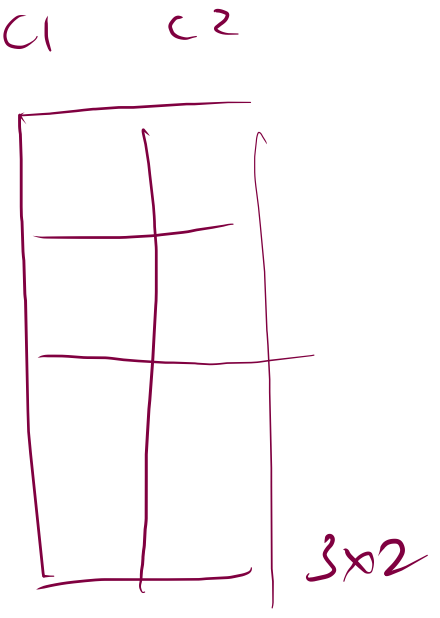
1,0 i<j 0

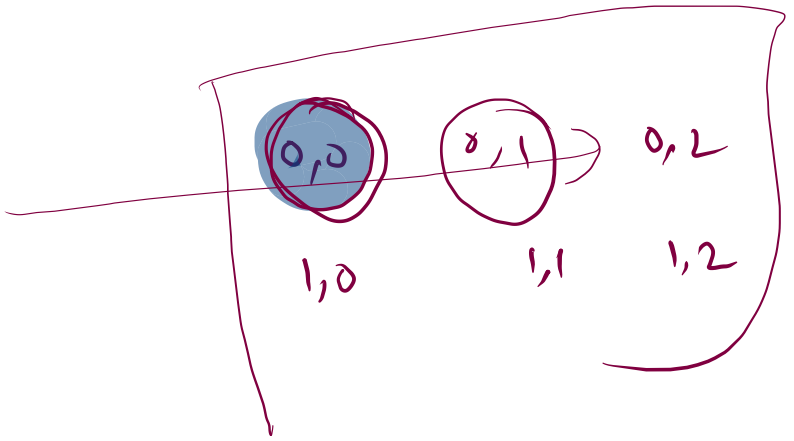


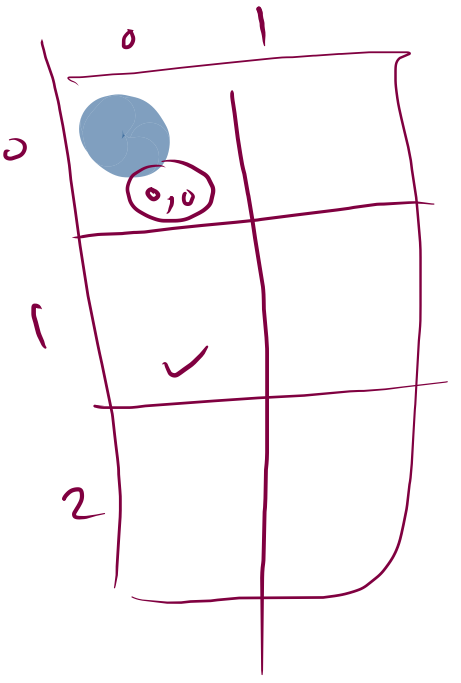


2 x 3





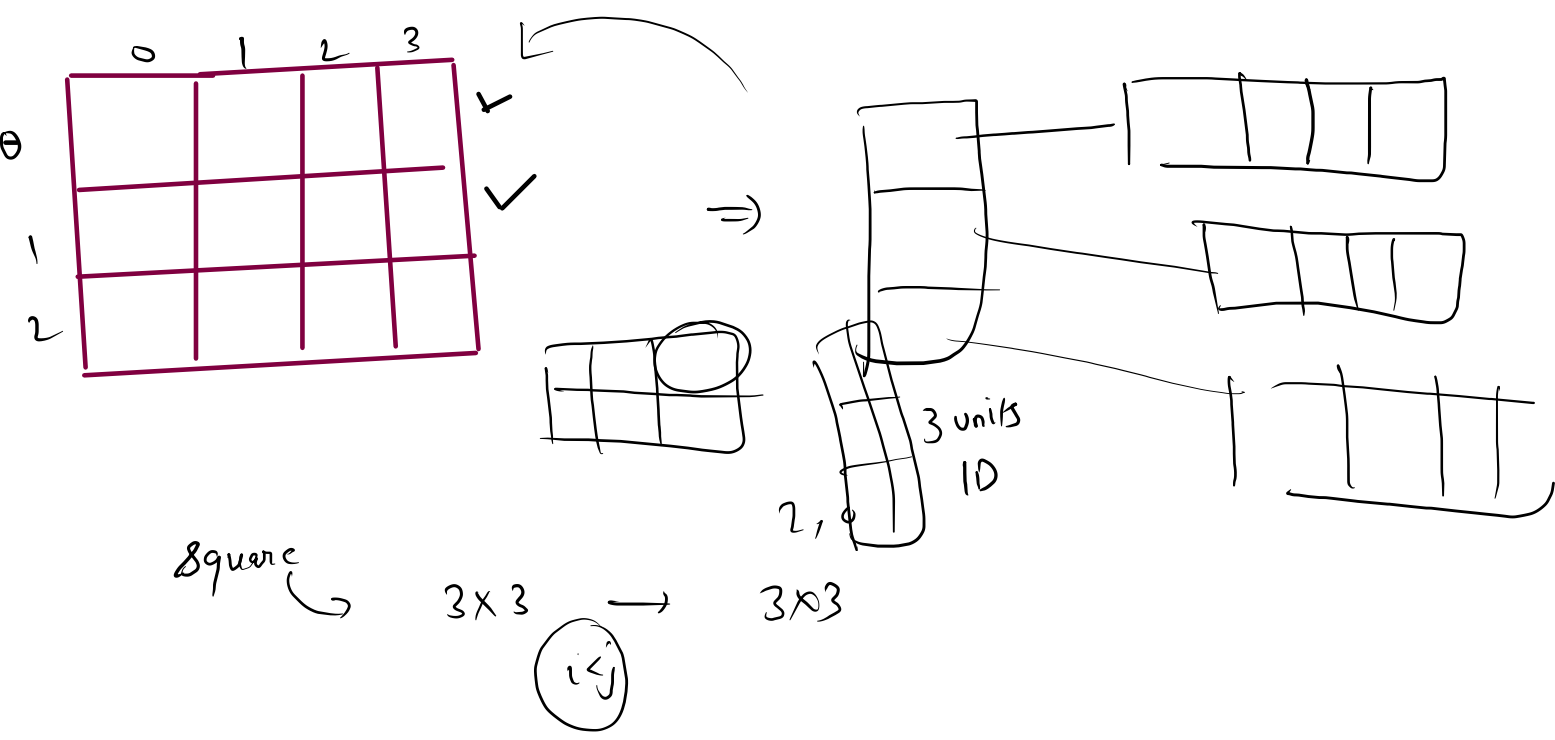




i j

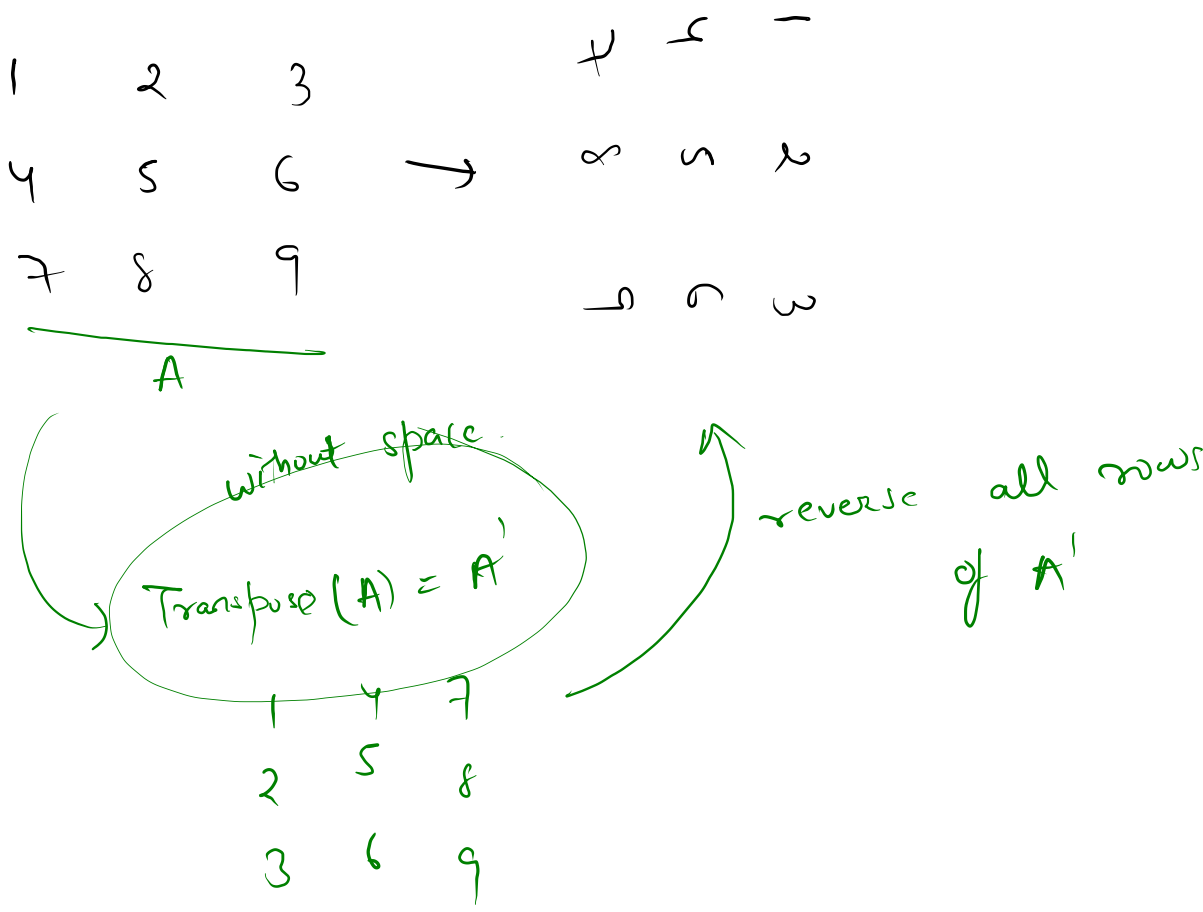
0 1

1 0


0 1 2 3
0
1
2
3 units
1D
2, 0
Square
3 x 3
3x3
i<j

48. Rotate Image

Medium 16326 713 Add to List Share



~ O(m+n)

1	3	5	7
10	11	16	20
23	30	34	60

(m x n) → O(m²)

x = 30

me = x ans.

me > x

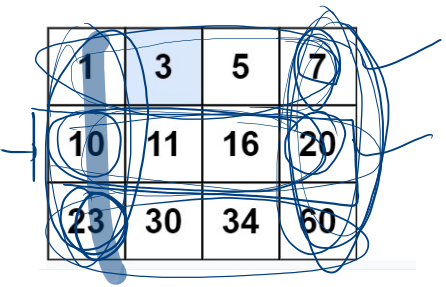
← left (j--)

me < x

↓ down

(i++)

me		x
7	<	30
20	<	30
60	>	30



left, down

x = 15

x = 70

7 < 15	
20 > 15	
16 > 15	
11 < 15	
30 > 15	
23 > 15	

log(n) + log(m)

O(log(n x m))

ceil

floor

4.7] = 5

4

16

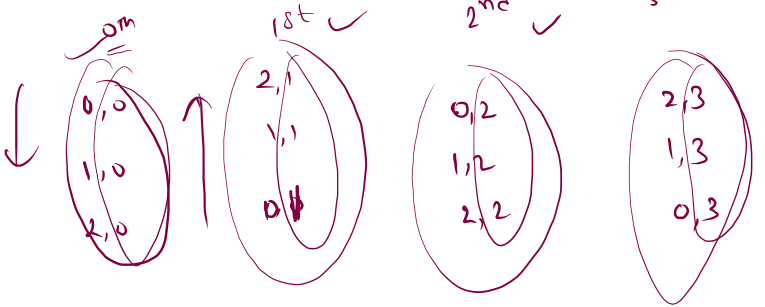
11

Wave Print

	0	1	2	3
0	1	3	5	7
1	10	11	16	20
2	23	30	34	60

1 10 23 30 11 3 ...

5 16 34 60 20 7 ✓



fix → outloop

for (j = 0 ⟶ j<n (cols))

{ if (j%2==0) ⟶ even

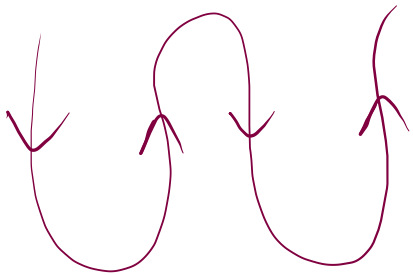
{ for (i=0 ⟶ i<n)

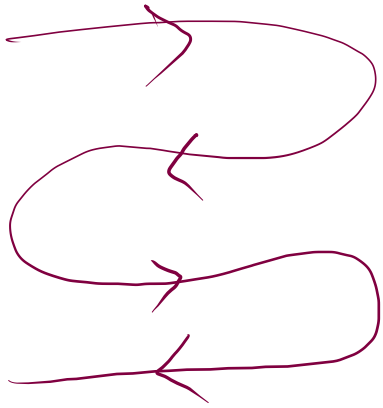
}

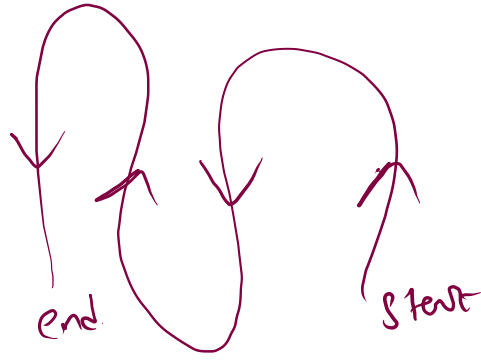
else { j is odd

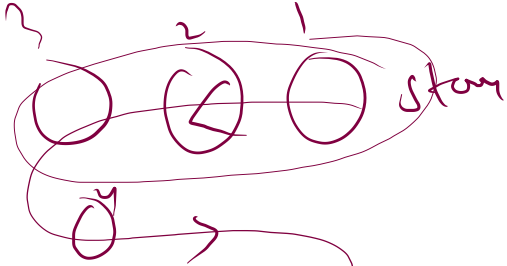
for (i=n ⟶ 0)

} }






end
start


2
1
start


end