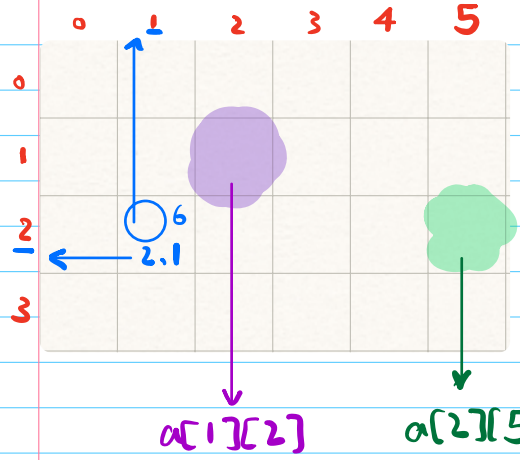
7805AM

Topics:- 2D array Intro ✓
- row wise, col wise print ✓
- diagonals ✓
- trasnpose
- rotate

2D Matrix [redacted]

declare: a[4][6] → a[row][col]



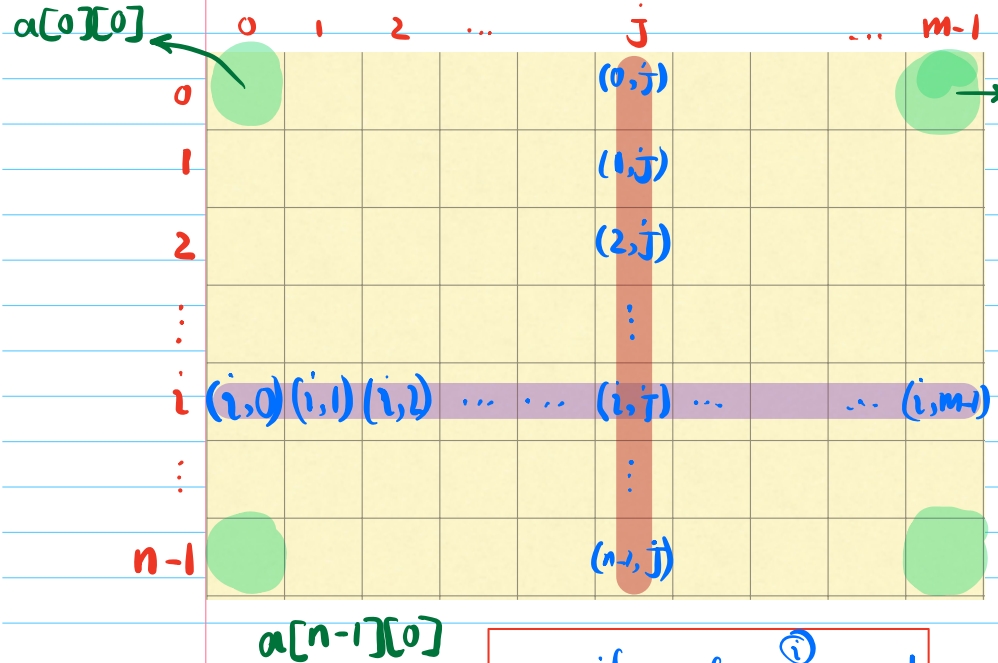
a[n][m]

rows [0, n-1]

cols [0, m-1]

obs1: if we fix column (j) and iterate over, rows move [0, n-1]

int a[n][m]



Quiz

Quiz

obs2: if we fix row (i) and iterate over, cols move [0, m-1]

P1 Given a[n][m] print row wise sum

n, m > 0

ex a[3][4]

	0	1	2	3
0	4	3	-1	7
1	9	2	3	1
2	2	2	3	0

i, j

output

13

15

7

```
void rowWiseSum(int a[][]){
    int N = a.length
    int M = a[0].length
    for(i=0; i<n; i++){
        // for each row
        sum=0
        for(j=0; j<m; j++){
            sum += a[i][j]
        }
        print(sum)
    }
}
```

Quiz

TC: O(n×m)

SC: O(1)

- Can you optimize ↑ furthure?

P1.2 Given a[N][M], print column wise sum.

↳ your assignment

ex a[3][4]

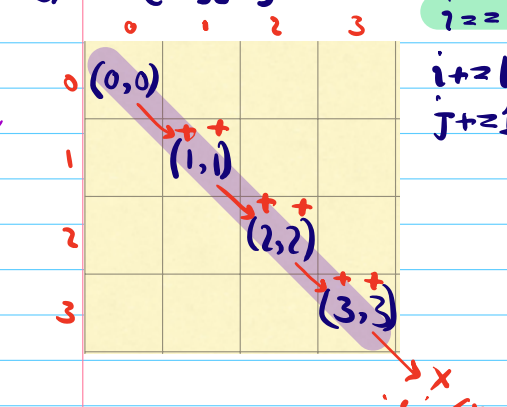
	0	1	2	3
0	4	3	-1	7
1	9	2	3	1
2	2	2	3	0
	15	7	5	8

Principal

P2 Given Square mat. a[N][N] print diagonals
- ① Leading (main)
- ② Trailing (anti)

top left, top right, bottom left, bottom right

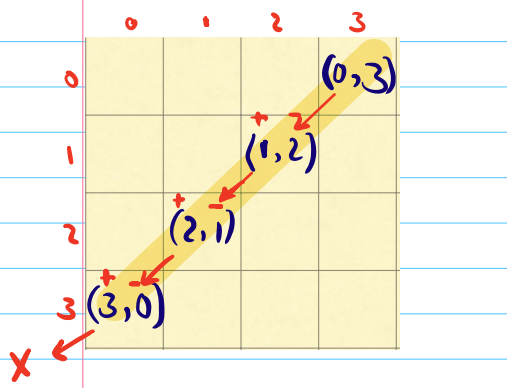
ex a[4][4]



main

i==j
i+=1
j+=1

Quiz
TC: O(n)
SC: O(1)




```
print diagonals(a[][])
// main
i=0
j=0
n = a.Len
while(i<n && j<n){
n   print(a[i][j]+" ")
    i+=1; j+=1
}
print("\n") // new line
// anti
i=0
j= n-1
while(i<n && j>=0){
n   print(a[i][j]+" ")
    i+=1; j-=1
}
}
```

O(n+n)

main

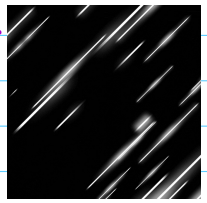
```
for(i=0;i<n;i++){
    print(a[i][i])
}
```

```
for(i=0;i<n;i++){
    print(a[i][(n-1)-i]
}
```

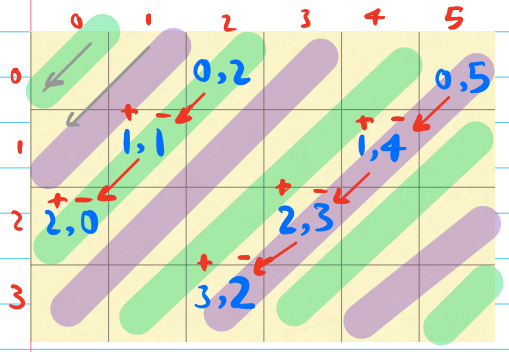
Anti

P3 Given a 2D array a[N][M] print all elements diagonaly right to left. i.e order → top row → right to left then last col top to bottom

r→L



ex a[4][6]



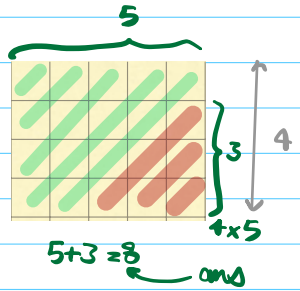
Quiz ✓

Quiz ✓

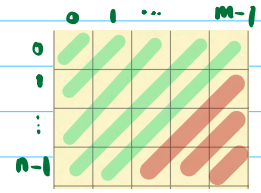
(*1) i++

(*2) J--

(*3) 0<=i && i<N
0<=J && J<M

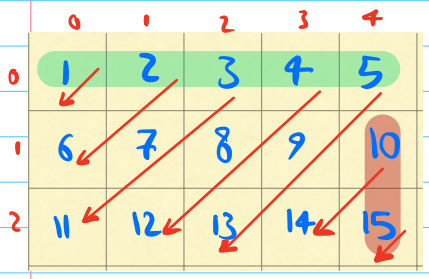


5+3 = 8 ← ans



ans → m + n - 1

strat ex a[3][5]



output

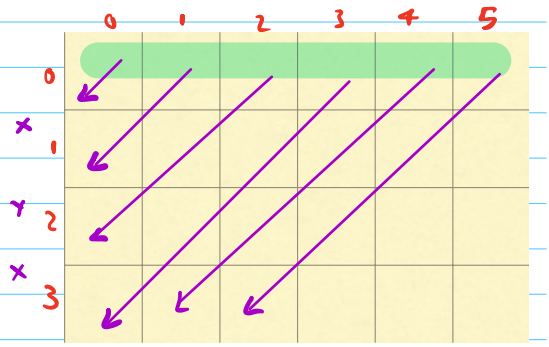
1
2 6
3 7 11
4 8 12
5 9 13
10 14
15

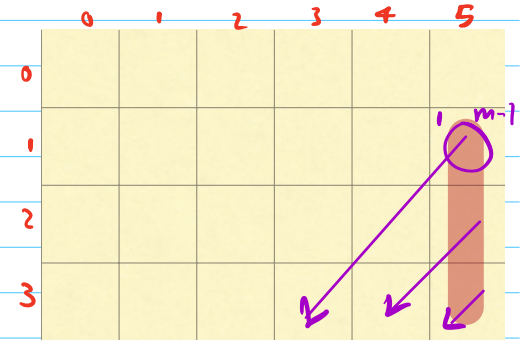
Quiz

TC:
O(N×M)

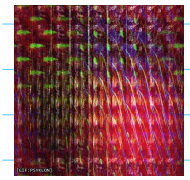
SC:
O(1)

```
void printRLDiagonals(int a[][]){
    int N = a.length
    int M = a[0].length
    for(c=0; c<m; c++){
      i=0; j=c
      while(i<n && j>=0)   *3
        print(a[i][j]+" ")
        i++; j--;   *1 *2
      }
      print("\n") //new line
    }
    for(r= 1; r< n ; r++){
        i=r; j=m-1
        while(i<n && j>=0)
          print(a[i][j]+" ")
          i++; j--;   *1 *2
        }
        print("\n") //new line
    }
}
```





P4 Given square mat. a[N][N] calculate transpose of a, with SC: O(1) ✓ .i.e transpose a in place & return



a[5][5]

(a) transpose: i^{th} row → i^{th} col

a^T, a'

	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

(*4) main diag. unchanged

(*5) i,j ↔ j,i

(*6) ignore the main diag

(*7) for loop only on either top or bottom △ & swap

```
int[][] transpose(int a[][]){
    int N = a.length
    for(i=0; i<n; i++){
        for(j=0; j<n; j++){
            // i,j
            temp = a[i][j]
            a[i][j] = a[j][i]
            a[j][i] = temp
        }
    }
    ret a
}
```

← assignment

fix bug

apply (*6) & (*7)

Quiz

Tc: $O(N^2)$

SC: $O(1)$

$O(N \times N)$

$O\left(\frac{N \times N}{2} - \frac{N}{2}\right) = O\left(\frac{1}{2}N^2 - \frac{N}{2}\right)$

what if a[][] is rectangle

ex a[4][6]

	0	1	2	3	4
0	1	2	3	4	5
1	6	7	8	9	10
2	11	12	13	14	15

in place is not possible if not square

a^T

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

P5 Given a 2D sqr. mat. a[n][n] rotate it clock wise from top right corner.

hint: use transpose

SC: O(1)

a

	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

a_{90}

	4	3	2	1	0
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

a^T

	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

a_{90}

	4	3	2	1	0
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

```
int[][] rotate90(a[][]){
    transpose(a) // SC O(1)
    for(i=0; i<n; i++){
        reverseArray(a[i]) SC O(1)
    }
    ret a;
}
```

n = a.len

TC $O(n^2)$ + $O(n^2)$

TC $O(n^2)$

SC $O(1)$