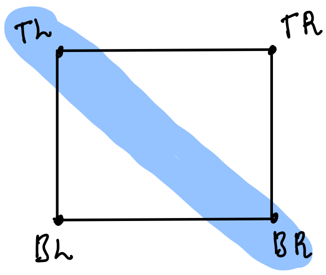
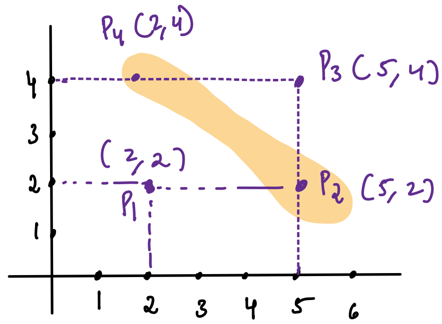
Todays Content:

- Sub matrix sum queries
- #no: of submatrices
- Sum of all submatrices
- Find k in a row-wise & column wise sorted

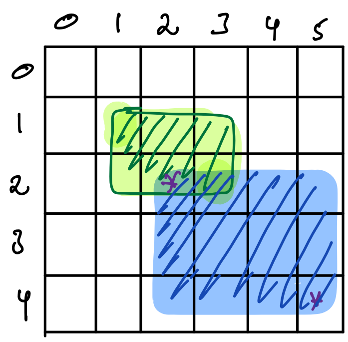
Sub Matrix:





Idea: Given 2 opp corners our submatrix fixed { $\boxed{\text{TL \& BR}}$ ✓ / BL & TR

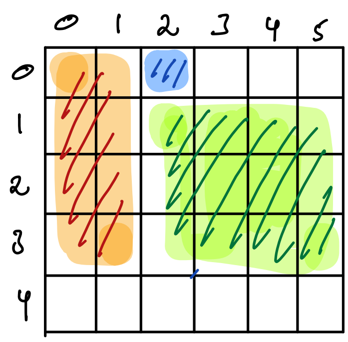
mat[5][6]



(x_1, y_1) (x_2, y_2)
TL & BR

(1,1) – (2,3) → 1st sub

(2,2) – (4,5) → 2nd sub



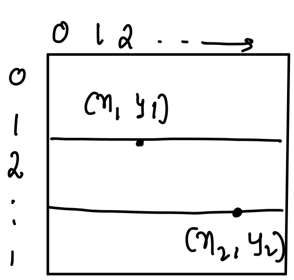
(x_1, y_1) (x_2, y_2)
TL & BR

(1,2) (3,5) → 12 ele

(0,0) (3,1) → 8 ele

(0,2) (0,2) → 1 ele

#obs: Submatrix can have only 1 row & only 1 col as well



obs: $x_2 >= x_1$ && $y_2 >= y_1$

(x_1, y_1) TL (x_2, y_2) BR

Given a mat[N][M] & Q queries $\left[\begin{matrix}(x_1, y_1) & & (x_2\ y_2) \\ TL & \& & BR\end{matrix}\right]$ $\underline{x_2 >= x_1\quad y_2 >= y_1}$

Find the submatrix sum for every query

Ex:

	0	1	2	3	4
0	7	2	1	0	3
1	3	1	2	3	9
2	6	3	2	4	1
3	3	1	2	1	4
4	2	-1	6	2	3
5	-1	3	2	1	4

Idea1: For every query, iterate in submatrix & get its sum.

TC: Q * [N*M]   SC: O(1)

Idea2: In ar[N], create pf[N]

pf[i] = sum of all ar[0] → ar[i]

In mat[N][M], create pfmat[N][M]

Pfm[i, j] = sum of all mat[0, 0] → mat[i, j]

Q = 2

TL & BR		Sum
(2, 1)	(4, 3) :	20
(3, 2)	(5, 4) :	25

Pfm [1,2] = mat[0 0] → mat[1,2]

Pfm [2, 1] = mat[0,0] → mat[2,1]

Pfm [1, 1] = mat[0,0] → mat[1,1]

mat[3][3]

	0	1	2
0	a	b	c
1	d	e	f
2	g	h	i

→ Prefix Sum in Row

a	a+b	a+b+c
d	d+e	d+e+f
g	g+h	g+h+i

→ Prefix sum in col

	0	1	2
0	a	a+b	a+b+c
1	a+d	a+b d+e	a+b+c d+e+f
2	a+d+g	a+b d+e g+h	a+b+c d+e+f g+h+i

	0	1	2	3
0	2	-1	3	1
1	4	2	3	5
2	-1	2	1	0

→ Pf row wise

2	1	4	5
4	6	9	14
-1	1	2	2

→ Pf col wise

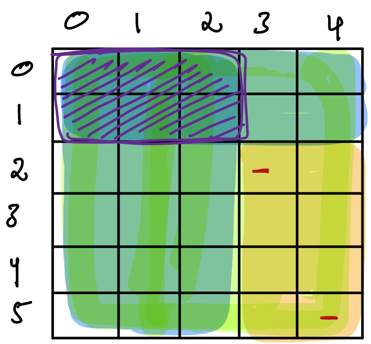
	0	1	2	3
0	2	1	4	5
1	6	7	13	19
2	5	8	15	21

Pfmat [1 2] = mat[0,0] → mat[1, 2] = 13 ✓

Pfmat [2, 3] = mat [0,0] → mat[2,3] = 21 ✓

Queries:

mat[6][5] → pfm[6][5]



Q: 2:

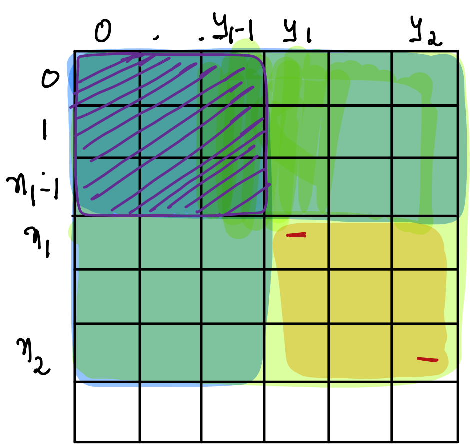
TL	BR
(2,2)	(4,4)

$S = pfm[4,4] - pfm[4,1] - pf[1,4] + pf[1,1]$

(2,3)	(5,4)

$S = pf[5,4] - pf[5,2] - pf[1,4] + pf[1,2]$

// Generalized Query:



TL BR

Q: $\{x_1, y_1\}$ $\{x_2\ y_2\}$

```
S = pfm[x2, y2]
if( y1 > 0 ){
    S = S - pfm[x2, y1-1]
}
if( x1 > 0 ){
    S = S - pfm[x1-1, y2]
}
if( x1 > 0 && y1 > 0 ){
    S = S + pfm[x1-1, y1-1]
}
```

void SumSub(int mat[N][M], int que[Q][4]){

que[i][0] x_1 $\quad$ x_2 que[i][2]
que[i][1] y_1 $\quad$ y_2 que[i][3]
$\underbrace{\text{que[i][0], que[i][1]}}_{TL}$ $\quad$ $\underbrace{\text{que[i][2], que[i][3]}}_{BR}$

int pfm[N][M]

Step 1: Copy mat[][] → pfm[][] : O(N*M)

Step2: On every row of pfm[][] apply pfsum : O(N*M)

Step3: On every col of pfm[][] apply pfsum : O(M*N)

We can interchange 2 & 3

i=0; i< Q; i++){

    x_1 = que[i][0] y_1 = que[i][1] → $\boxed{\{TL\}}$
    x_2 = que[i][2], y_2 = que[i][3] → $\boxed{\{BR\}}$
    // Sum of submatrix (x_1 y_1) → (x_2 y_2)
    S = pfm[x_2 y_2]
    if(y_1 > 0){
        S = S - pfm[x_2, y_1-1]
    }
    if(x_1 > 0){
        S = S - pfm[x_1-1, y_2]
    }
    if(x_1>0 && y_1>0){
        S = S + pfm[x_1-1, y_1-1]
    }
    print(s) // printing submatrix sum query
}
}

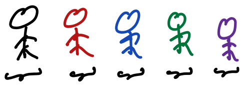
[TC: O(N*M) + Q*O(1)]

TC: $O(\underbrace{N*M}_{\text{copy}} + \underbrace{N*M}_{\text{row pf[]}} + \underbrace{N*M}_{\text{col pf}} + \underbrace{Q}_{\text{Q queries}} * \underbrace{1}_{\text{TC for each query}})$

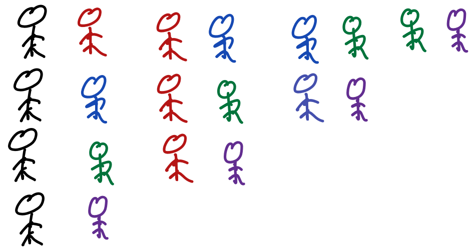
copy → won't be there if inplace

SC: O(N*M) : pfmat[][]
SC: O(1) : [If we apply row wise & column wise pfsum in mat[][]]

Q) 5 diff peop how many ways we can pick 2 people ?



$4 + 3 + 2 + 1 + 0 = 10$



Q) Given N, how many ways we can pick 2 $= \frac{(N)(N-1)}{2}$

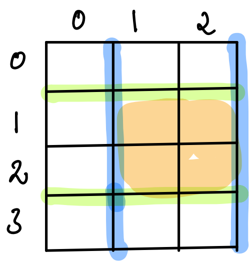
$a_1 \quad a_2 \quad a_3 \quad \ldots \quad a_N$

$N-1 \quad N-2 \quad N-3 \quad \ldots \quad 0$

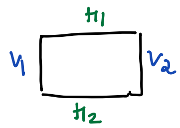
Total pairs

$N-1 + N-2 + N-3 + \ldots 0 = \frac{(N)(N-1)}{2}$

Q) mat[4][3] how many submatrix are there



submatrix → Hori lin 2, Vertical 2



HLines : 5 * Vertical : 4

10 * 6 = 60 Submatr

obs: Fix 2 horizontal lines as boundaries &
fix 2 vertical lines as boundaries
Submatrix fixed

Q) mat[N][M] how many subMtrix are there?

Horizontal = (N+1) # ways to select 2 = $\frac{(N+1)(N)}{2}$

$*$

Vertical = (M+1) # ways to select 2 = $\frac{(M+1)(M)}{2}$

Total Submatrix $= \frac{(N+1)(N)\,(M+1)(M)}{4}$

2Q) Given mat[N][M] find sum of all submatrix sums:

↳ Similar: Sum of every subarray sum

Idea: Contribution technique

: $\sum ar[i] * c_i$ → { How many time ar[i] come in subarrays }

: $\sum mat[i][j] * c_{ij}$ → { In how many submatrix mat[i][j] comes }

Ex:

	0	1	2	3	4
0	TL	TL	TL		
1	TL	TL	TL		
2	TL	TL	TL BR	BR	BR
3			BR	BR	BR

Submatrix

TL	& BR
0 0	2 2
0 1	2 3
0 2	2 4
1 0	3 2
1 1	3 3
1 2	3 4
2 0	
2 1	
2 2	

Count of submat

TL * BR   Submatrix

9   6   => 54

Ex2:

	0	1	2	3
0	TL	TL	TL	
1	TL	TL	TL BR	BR
2			BR	BR

Submatrix

TL & BR   Submatrix

6   4   24

Generalize: # No: of submatrix all [i, j]

$\underbrace{0 \ldots j}_{j+1}$ $\underbrace{j \ldots M-1}_{M-j}$

	0	.	j			M-1
0	TL	TL	TL			
1	TL	TL	TL			
:	TL	TL	TL			
i	TL	TL	BR	BR	BR	BR
			BR	BR	BR	BR
			BR	BR	BR	BR
N-1			BR	BR	BR	BR

Rows 0 … i: $i+1$; rows i … N-1: $N-i$

Submatrix

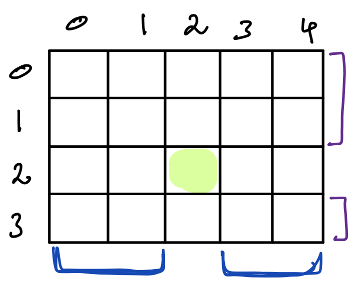
TL & BR

TL: $(i+1)(j+1)$

BR: $(N-i)(M-j)$

No: of sub in which mat[i, j] present = $(i+1)(j+1)(N-i)(M-j)$

Ex:



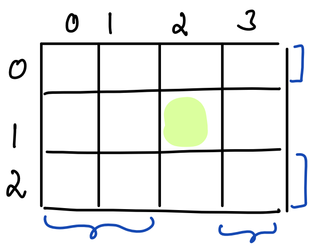
Submatrix

#R: 3*2 = 6

VR = 3*3 = 9

= #R * VR = 9*6 = 54

Ex2:



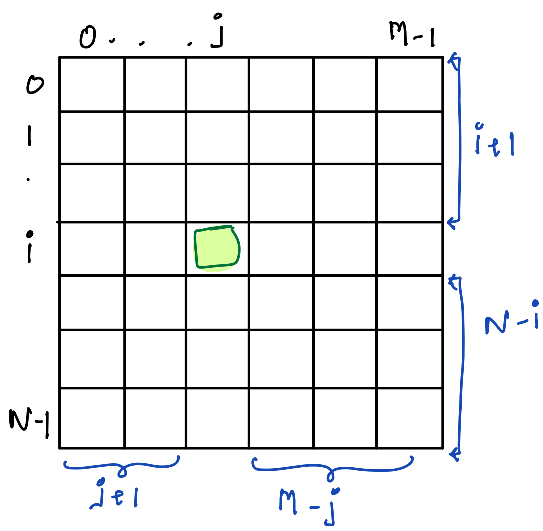
Submatrix

#R: 2*2 = 4

VR: 3*2 = 6

= 24

Generalize: # No: of submatrix cell [i, j]



Submatrix

#R: (i+1)(N-i)

VL: (j+1)(M-j)

#R*VL = (i+1)(N-i)(j+1)(M-j)

3Q) Given row wise column wise sorted matrix, find max submatrix sum = ?

TODO

assignment:

Ex1:

	0	1	2	3
0	-20	-16	-10	-7
1	-15	-9	4	6
2	-10	2	7	9
3	-3	7	10	12

Ex2:

	0	1	2	3
0	-20	-16	-4	-1
1	-10	-8	-2	5
2	-4	2	4	8

Ex3:

	0	1	2
0	-50	-40	-30
1	-35	-20	-15
2	-19	-14	-3

```
int    maxSum (int mat[N][M]) {

}
```

4Q) Given row wise, column wise sorted matrix search k?

TL, TR, BL, BR; k = 12

	0	1	2	3	4	5
0	-10	-5	-2	2	4	7
1	-7	-4	-1	3	6	9<12
2	-2	3	5	7	11<12	14>12
3	6	6	8	11<12	14>12	17
4	7	11	12	15>12	19	20
5	13	14	18	20	24	29

Idea:

1) Search entire matrix
   TC: $O(N*M)$ SC: $O(1)$

2) For every row compare extreme with k, if k can be present in row search in row
   TC: $O(N*M)$ SC: $O(1)$

k = 18

	0	1	2	3	4	5
0	-10	-5	-2	2	4	7<18
1	-7	-4	-1	3	6	9<18
2	-2	3	5	7	12	14<18
3	6	6	8	11	14	17<18
4	7	11	12	15	19>18	20>18
5	10	14	18	20	24	29

```
bool search(int mat[N][M], int k){
    i=0, j= m-1
    while( i<n && j>=0 ){
        if( mat[i][j] < k){ //skip row
            i++
        }
        else if (mat[i][j] > k){ //skip col
            j--
        }
        else { return True }
    }
    return False
}
```

TODO: TL TR ✓ BL BR

Check if we can do it from TL, BL, BR

TC: At every step we skip a row or a col
→ Total Steps: (N+M)
→ SC: O(1)