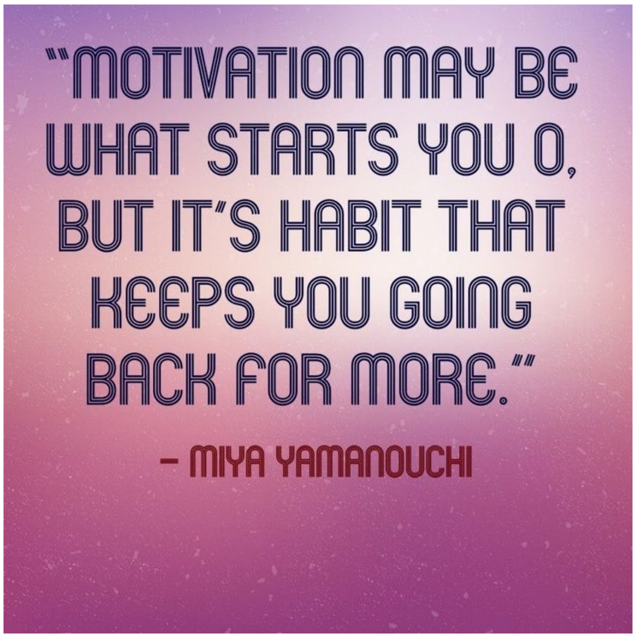
ARRAYS 2

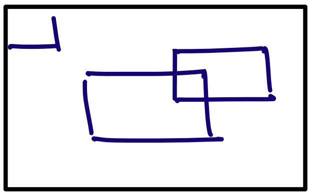


Good
Evening

Today's Content

01. Introduction to submatrix
02. Submatrix sum queries
03. Sum of all submatrix sum
04. Maximum sum submatrix { Hint } for sorted matrix

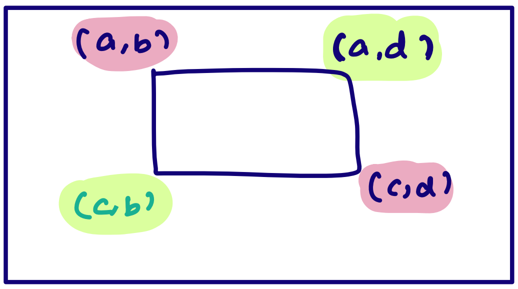
Submatrix : Part of a matrix is a submatrix



Note :- (a) Single ele is a submatrix

(b) Complete matrix is also a submatrix

Represent a submatrix →



No need for the 4 corners

TL & BR

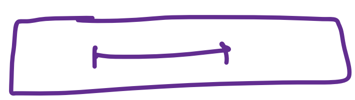
TR & BL

int mat[5][6];

	0	1	2	3	4	5
0	2	-1	3	2	1	3
1	3	2	6	2	6	7
2	10	9	8	2	2	1
3	4	-1	2	3	4	2
4	3	2	6	9	8	9

TL = (2,1) (2,3)

BR = (4,1 (4,3)



Q Given a matrix of size $N * M$ & Q queries

For each query, find sum of given submatrix

Note :- TL is top left & BR is bottom right.

	0	1	2	3
0	2	-1	3	2
1	3	2	6	2
2	10	9	8	2
3	4	-1	2	3
4	3	2	6	9

Queries = 2

(T, L) (B, R)

01. (2,1) & (4,2) → 26

02. (1,1) & (3,3) → 33

Brute force → For every query, we need to iterate from TL to BR & get the sum.

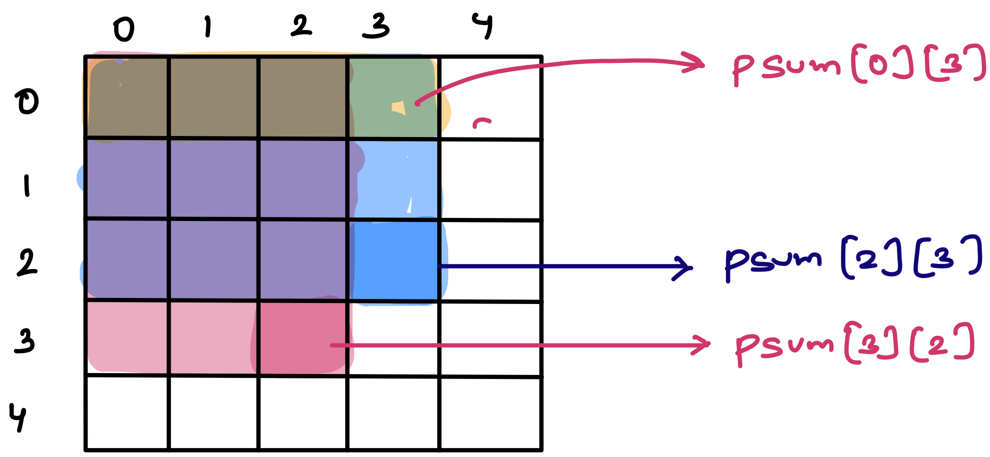
TC : $O(Q * N * M)$

SC : $O(1)$

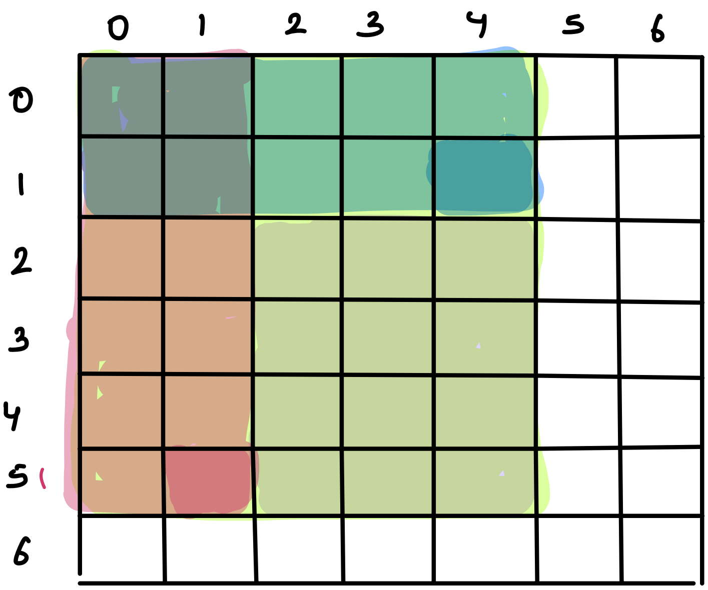
Idea 2 → Psum approach

1 D array psum[i] → sum of all elements from 0 to i

psum[i][j] → sum of all elements from (0,0) to (i,j) psum[4][4] =



* Assume psum matrix

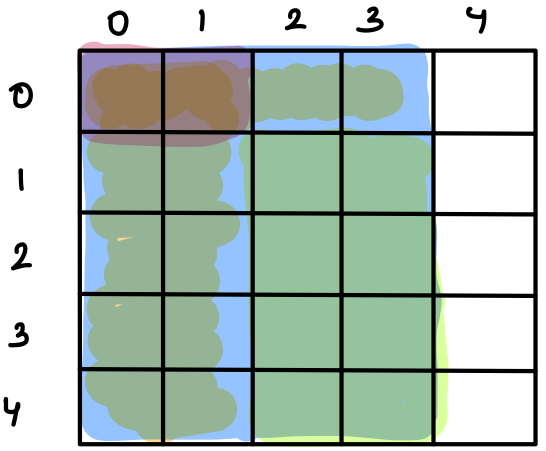


psum matrix

Query

(2,2) → (5,4)

⇒ sum(5,4) − sum(5,1) − sum(1,4) + sum(1,1)

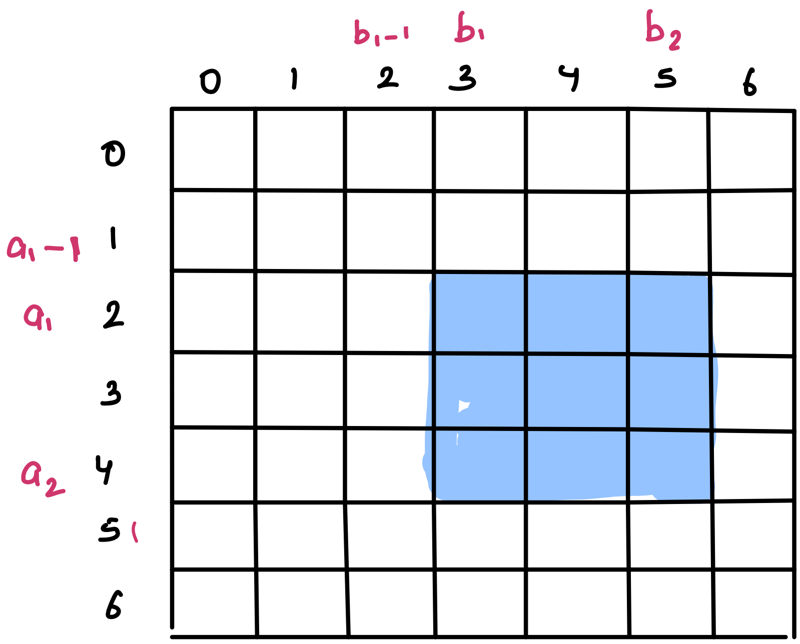


psum[ ][ ]

Query

$(1,2) \longrightarrow (4,3)$

$$psum(4,3) - psum(4,1) - psum(0,3) + psum(0,1)$$



T L

(a_1, b_1)

B R

$(a_2\ b_2)$

$$psum(a_2, b_2) - psum(a_2, b_1-1) - psum(a_1-1, b_2) + psum(a_1-1, b_1-1)$$

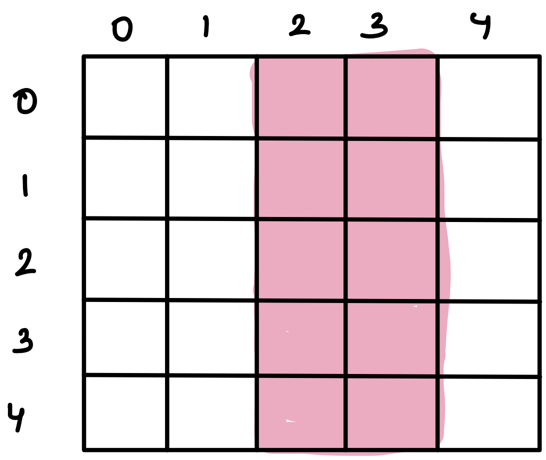
Query = a_1, b_1 to a_2, b_2

$ans = psum(a_2, b_2)$

if $(b_1 > 0)$ { $ans = ans - psum(a_2, b_1 - 1)$

if $(a_1 > 0)$ { $ans = ans - psum(a_1 - 1, b_2)$

if $(a_1 > 0$ && $b_1 > 0)$ { $ans = ans + psum(a_1 - 1, b_1 - 1)$



Query

$(0, 2) \longrightarrow (4, 3)$

$psum(4, 3) - psum(4, 1)$

* **Build psum array**

01. Apply psum in rows.

02 Apply psum in cols.

	0	1	2
0	a_0	b_0	c_0
1	a_1	b_1	c_1
2	a_2	b_2	c_2

arr

→

	0	1	2
0	a_0	a_0+b_0	$a_0+b_0+c_0$
1	$a_0 + a_1$	$a_0+b_0 + a_1+b_1$	$a_0+b_0+c_0 + a_1+b_1+c_1$
2	$a_0 + a_1 + a_2$	$a_0+b_0 + a_1+b_1 + a_2+b_2$	$a_0+b_0+c_0 + a_1+b_1+c_1 + a_2+b_2+c_2$

psum

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

mat

$\xrightarrow[\text{psum mat}]{\text{final}}$

	0	1	2	3
0	3	5	9	10
1	2	8	15	18
2	4	17	30	36

psum

	0	1	2	3
→ 0	3	2	4	1
→ 1	-1	4	3	2
→ 2	2	7	6	3

$\xrightarrow[\text{on rows}]{\text{Apply psum}}$

	0	1	2	3
0	3	5	9	10
1	-1	3	6	8
2	2	9	15	18

d

Apply psum
on cols

$$\begin{array}{c|cccc} & 0 & 1 & 2 & 3 \\ \hline 0 & 3 & 5 & 9 & 10 \\ 1 & 2 & 8 & 15 & 18 \\ 2 & 4 & 17 & 30 & 36 \end{array}$$

* Using psum matrix logic

$$T.C = O(\underbrace{N * M + Q}_{\text{Queries}})$$

$S.C = O(N * M) \longrightarrow$ To make psum matrix

O(1) if we are modifying the same matrix

Q2. Given a matrix of size N * M. Calculate sum of all submatrix sums

$$\begin{bmatrix} 3 & 1 \\ -1 & -2 \\ 2 & 4 \end{bmatrix} = \left\{ \begin{matrix} [3] & [3, 1] & \begin{bmatrix} 3 & 1 \\ -1 & -2 \end{bmatrix} & \begin{bmatrix} 3 & 1 \\ -1 & -2 \\ 2 & 4 \end{bmatrix} & \begin{bmatrix} 3 \\ -1 \end{bmatrix} & \begin{bmatrix} 3 \\ -1 \\ 2 \end{bmatrix} \\ [1] & \begin{bmatrix} 1 \\ -2 \end{bmatrix} & \begin{bmatrix} 1 \\ -2 \\ 4 \end{bmatrix} & [-1] & [-1, -2] & \begin{bmatrix} -1 & -2 \\ 2 & 4 \end{bmatrix} \\ \begin{bmatrix} -1 \\ 2 \end{bmatrix} & [2] & [2, 4] & [4] & [-2] & \begin{bmatrix} -2 \\ 4 \end{bmatrix} \end{matrix} \right.$$

sum = 36

BF → Consider all the submatrices, get the sum & add it to the total sum

To represent a submatrix, TL & BR

(0,0)

↓

2 nested loop

	0	1	2	3	4
0					
1					
2					
3					
4					

$TC: O(n^2 m^2)$

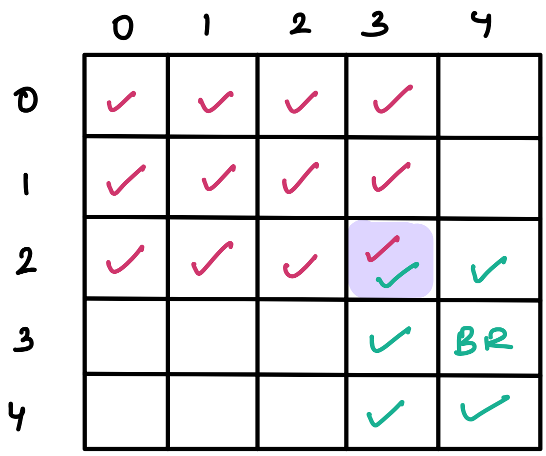
$SC: O(n * m)$

```
for ( a1=0 : a1<n : a1 ++)
    for (b1=0; b1<m ; b1 ++)
        TL = a1 , b1
        for (a2 = a1 ; a2<n ; a2++){
            for (b2 =b1 ; b2<m ; b2 ++){
                TL = a1 , b1     BR = a2 , b2
                sum = psum formula
            }
        }
    }
}
```

Contribution Technique =

ar[i]	*	contri		
3	*	6	=	18
1	*	6	=	6
-1	*	8	=	-8
-2	*	8	=	-16
2	*	6	=	12
4	*	6	=	24
				36

* How many no. of times is (2, 3) going to contribute

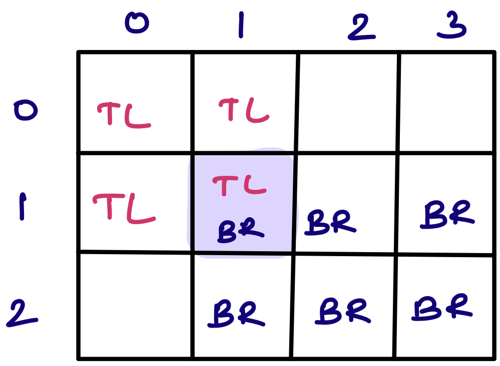


(2, 3)

Possible options for

TL		BR
12	*	6

= $\underline{72}$



i, j

(1, 1)

Possible options for

$\underline{\text{TL}}$		BR
4	*	6

= 24

$$\text{Generalise} = \underbrace{(i+1)*(j+1)}_{TL} * \underbrace{(n-i)*(m-j)}_{BR}$$

```
ans = 0

for ( i=0 ; i<n ; i++ )
    for ( j=0; j<m; j++ )
        contibution = (i+1) * (j+1) * (n-i) * (m-j)

        ans = ans + contibution * ar[i][j]
    }
}
```

TC : O(n * m)

SC : O(1)

03. Given row wise & col wise sorted matrix, find maximum submatrix sum

	0	1	2	3
0	-20	-16	-4	8
1	-10	-8	2	14
2	-1	6	21	30
3	5	7	28	42

sorted 1-D Array

maximum subarr sum

Claim → Include mat[n-1][m-1] in my ans

	0	1	2
0	-15	-14	-13
1	-12	-11	-7
2	-10	-9	-3

BR = [n-1][m-1]

→ Explore all TL corners

$(0,0)$

	0	1	2
0	✓	✓	✓
1	✓	✓	✓
2	✓	✓	✓

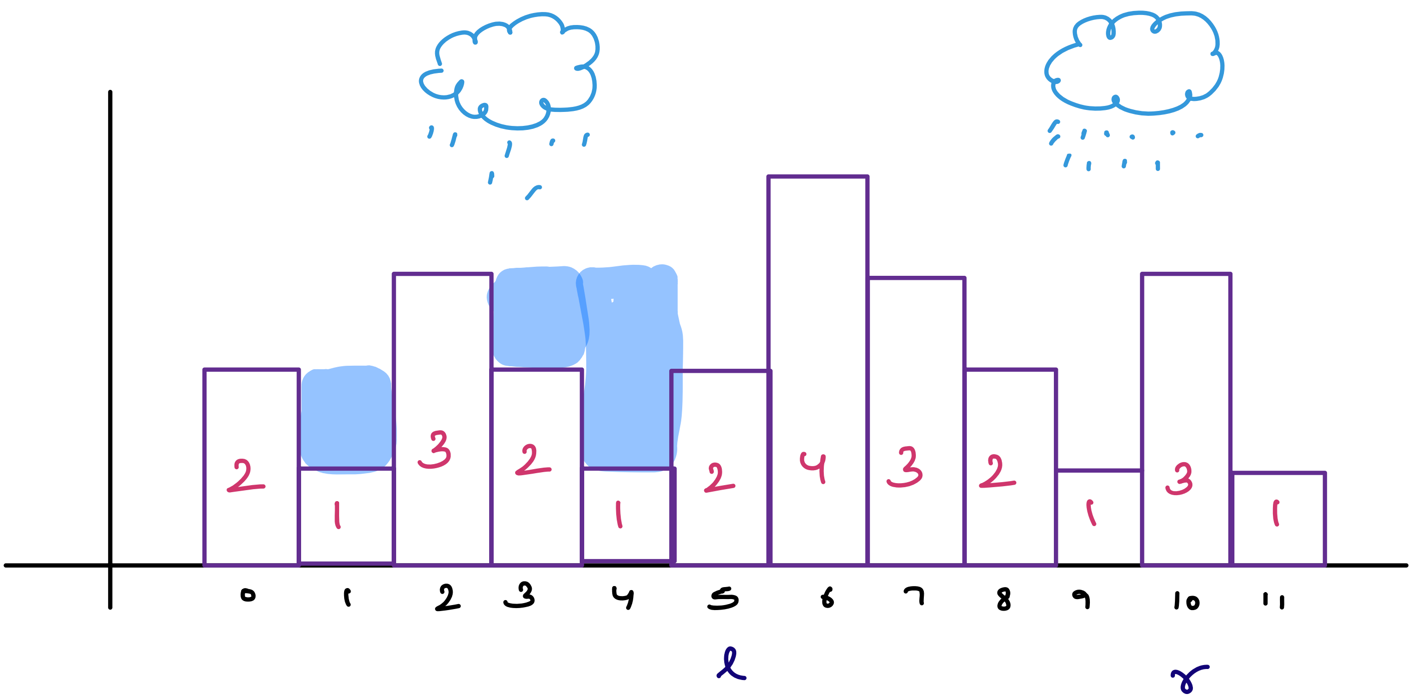
$3 * 3$

$n-1$ $\qquad\qquad$ $m-1$

$\underline{3-1} - i$ $\qquad\qquad$ $3-1$

$n-1 - i + 1$

$$\underbrace{2 - 0 + 1}_{3}$$



water = 3 − 1 = 2

lmax = 3

rmax = 3

resultat = 1 1 $\underbrace{1\ 1}$ 0

resultat = 1 1 $\underbrace{1\ 1\ 1}$

$\underbrace{1\ 1}$ 0 0 0

resultat 1 0 1 0 0

1 1 0 0 0

−1 −1 1 1 1