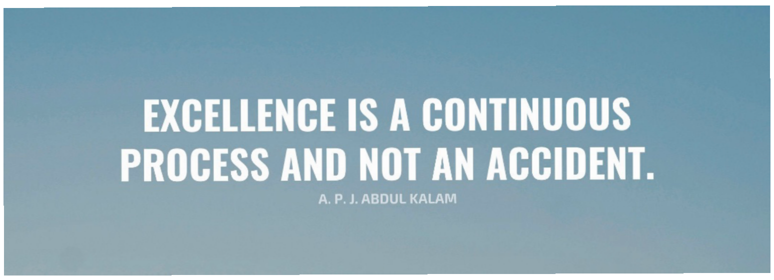
Good Evening Everyone !



Todays Content

- → Sub-matrix sum queries
- → Sum of all sub-matrices
- → Max submatrix sum.
- → Find an element in row-wise & column-wise sorted matrix.

psum[i] → Sum of all elements from idx 0 to idx i.

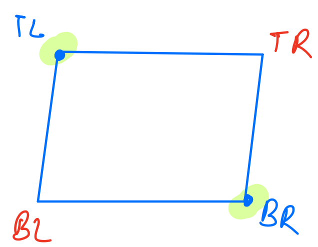
psum[i][j] → Sum of all elements from (0,0) to (i, j).

Q) Given a matrix of size $N * M$. For each query, find sum of given sub-matrix. → contiguous part of matrix.

→ whole matrix / singe cell → submatrix

Eg:

	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



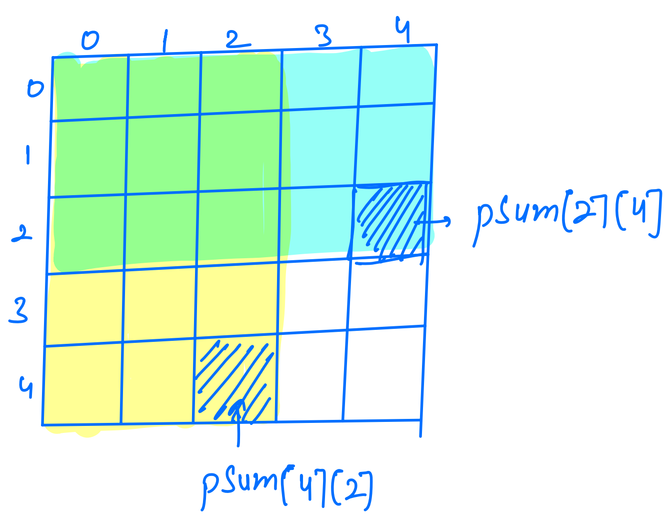
if TL and BR are fixed or TR and BL are fixed.
Then, we can uniquely identify sub-matrix.

Queries.

	T, L	B, R	
①	2, 1	4, 2	ans = 26
②	1, 1	3, 3	ans = 33

idea 1 → For every query, we need to iterate on sub-matrix and calculate the sum.

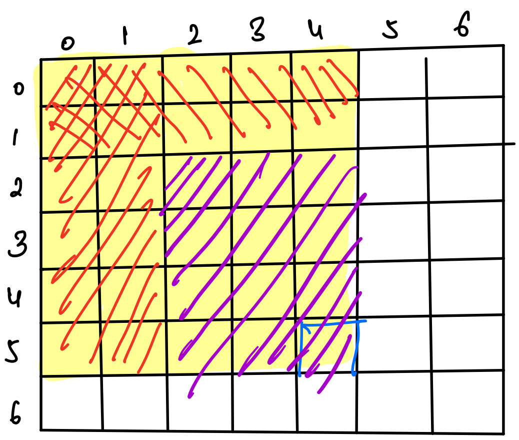
$T.C \to O(Q * N * M)$, $S.C \to O(1)$



idea 2 →

use psum[ ][ ];

// Assume pSum[][] ✓

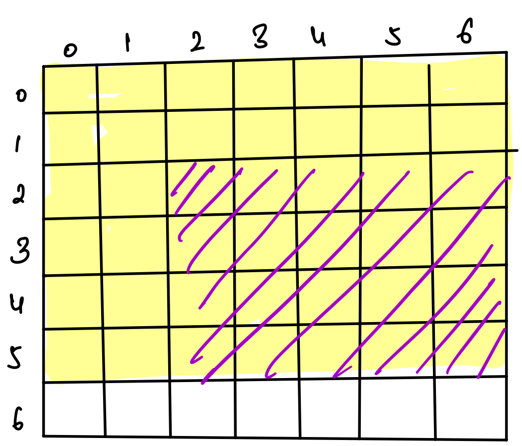


T.L   B.R

Query [2, 2   5, 4]

$pSum[5][4] - pSum[5][1] - pSum[1][4]$
$+ pSum[1][1]$

pSum[][].



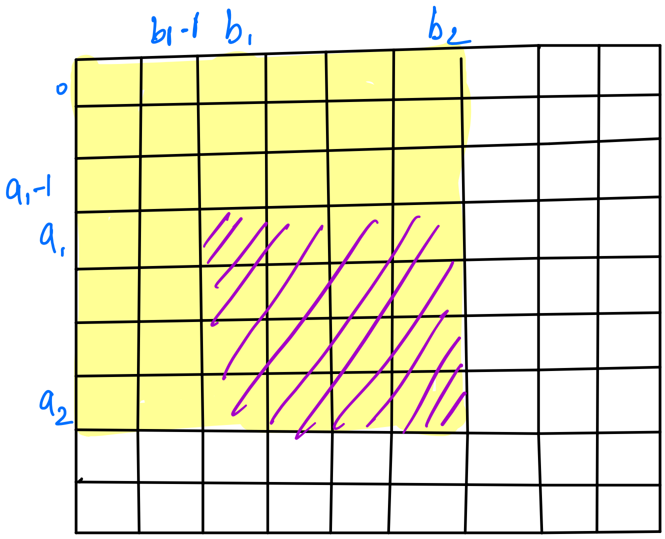
Query.

TL   BR

(2,2)   (5,6)

$pSum[5][6] - pSum[5][1] - pSum[1][6]$
$+ pSum[1][1]$

// Generalisation



TL: (a_1, b_1)

BR: (a_2, b_2)

$$pSum[a_2][b_2] - pSum[a_2][b_1 - 1] - pSum[a_1 - 1][b_2] + pSum[a_1 - 1][b_1 - 1]$$

pSum[][]

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

→ Given matrix

$pSum[1][2] \Rightarrow a_0 + b_0 + c_0 + a_1 + b_1 + c_1$

$pSum[2][1] \Rightarrow a_0 + b_0 + a_1 + b_1 + a_2 + b_2$

Apply pSum on every row ↓

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

Apply pSum on every column →

	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	$a_0+a_1+a_2$	$a_0+b_0+a_1+b_1+a_2+b_2$	$a_0+b_0+c_0+a_1+b_1+c_1+a_2+b_2+c_2$

$1 \le N, M \le 10^3$

$-10^5 \le A[i][j] \le 10^5$

Queries

T	L	B	R
2	2	5	6
1	0	3	4
0	0	3	5
3	4	5	5

code

```
long pSum[N][m];
for( i=0; i<N; i++){
    for( j=0; j<m; j++){
        pSum[i][j] = arr[i][j];
    }
}

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

Apply pSum on every row.

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

Apply pSum on every column.

```
for( i=0; i < Q.length; i++){
    a1 = Q[i][0],   b1 = Q[i][1]
    a2 = Q[i][2],   b2 = Q[i][3]
```

$$pSum[a_2][b_2] - \underbrace{pSum[a_2][b_1-1]}_{b_1>0} - \underbrace{pSum[a_1-1][b_2]}_{a_1>0} + \underbrace{pSum[a_1-1][b_1-1]}_{a_1>0 \,\&\&\, b_1>0}$$

```
}
```

$$\boxed{\begin{aligned} T.C &\to O(Q+N\cdot m) \\ S.C &\to O(N\cdot m) \end{aligned}}$$

arr

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

pSum

	0	1	2
0	2	5	9
1	7	17	24
2	16	24	47

Queries.

	T	L	B	R
1	1	1	2	2

Q) Given a matrix of size $N * M$. Calculate sum of all submatrix sums.

$\Rightarrow$ contribution technique

Eg →

$$\begin{array}{c|c} & \begin{array}{cc} 0 & 1 \end{array} \\ \begin{array}{c} 0 \\ 1 \\ 2 \end{array} & \begin{bmatrix} 3 & 1 \\ -1 & -2 \\ 2 & 4 \end{bmatrix} \end{array} =$$

$$[3] \quad [3,1] \quad \begin{bmatrix} 3 \\ -1 \end{bmatrix} \quad \begin{bmatrix} 3 & 1 \\ -1 & -2 \end{bmatrix} \quad \begin{bmatrix} 3 \\ -1 \\ 2 \end{bmatrix}$$

$$\begin{bmatrix} 3 & 1 \\ -1 & -2 \\ 2 & 4 \end{bmatrix} \quad [1] \quad \begin{bmatrix} 1 \\ -2 \end{bmatrix} \quad \begin{bmatrix} 1 \\ -2 \\ 4 \end{bmatrix} \quad [-1] \quad [-1 \;\; -2]$$

$$\begin{bmatrix} -1 \\ 2 \end{bmatrix} \quad \begin{bmatrix} -1 & -2 \\ 2 & 4 \end{bmatrix} \quad [-2] \quad \begin{bmatrix} -2 \\ 4 \end{bmatrix} \quad [2] \quad [2,4]$$

$$[4]$$

element		count
3	*	6
1	*	6
-1	*	8
-2	*	8
2	*	6
4	*	6
		36

B.F. → Consider all sub-matrices and for every submatrix calculate the sum.

$$\frac{N(N+1) \cdot m(m+1)}{4} \qquad T.C \to O(N^2 m^2)$$

idea. → Contribution technique

For every element, find how many times it is present in all the sub-matrices

idea :

In how many sub-matrices element at [2,3] will be present?

	0	1	2	3	4	5
0	TL	TL	TL	TL		
1	TL	TL	TL	TL		
2	TL	TL	TL	TL BR	BR	BR
3				BR	BR	BR
4				BR	BR	BR

5*6.

to define a sub-matrix TL. BR. are required.

TL. → 12 , BR. → 9

12 * 9 = 108.

[no. of options for T.L.] [no. of options for B.R]

Quiz

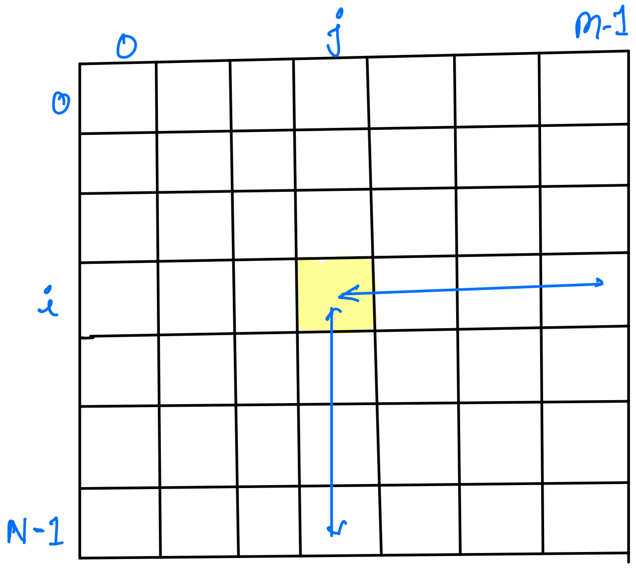
	0	1	2	3	4
0	TL	TL	TL		
1	TL	TL	TL BR	BR	BR
2			BR	BR	BR
3			BR	BR	BR

(1,2)

T.L B.R

6 * 9 = 54

// Given mat[N][M], in how many sub-matrices cell [i, j] is present?



options for T.L. $\downarrow$ $(i+1) * (j+1)$ * options for B.R. $\downarrow$ $(N-i) * (m-j)$

$[i, N-1]$

$N-\cancel{1}-i+\cancel{1}$

$\Rightarrow N-i$

$[j, m-1]$

$m-\cancel{1}-j+\cancel{1}$

$\Rightarrow m-j$

Code.

```
long ans = 0
for( i = 0; i < N; i++) {
    for( j = 0; j < m; j++){
        ans += arr[i][j] * (i+1) * (j+1) * (N-i) * (m-j)
    }
}
return ans;
```

T.C → O(NM)
S.C → O(1)

Maximum Sub-Matrix Sum

Given a 'row-wise and column-wise sorted matrix A of size N*M. Find maximum sub-matrix sum.

arr[][]

$$\begin{bmatrix} -10 & -6 & -2 & 1 \\ -5 & -4 & 4 & 5 \\ -3 & 0 & 6 & 20 \\ 2 & 5 & 10 & 25 \end{bmatrix}$$

25 → Always be a part of our ans.

idea → use sSum[][], where sSum[i][j] = Sum of all the elements from i,j to n-1, m-1

$$\begin{bmatrix} -10 & -6 & -2 & 1 \\ -5 & -4 & 4 & 5 \\ -3 & 0 & 6 & 20 \\ 2 & 5 & 10 & 25 \end{bmatrix}$$

↓ Apply sSum on every row

$$\begin{bmatrix} -17 & -7 & -1 & 1 \\ 0 & 5 & 9 & 5 \\ 23 & 26 & 26 & 20 \\ 42 & 40 & 35 & 25 \end{bmatrix} \xrightarrow[\text{on every column}]{\text{Apply sSum}} \begin{bmatrix} 48 & 64 & 69 & 51 \\ 65 & 71 & 70 & 50 \\ 65 & 66 & 61 & 45 \\ 42 & 40 & 35 & 25 \end{bmatrix}$$

ans.

code:→

```
long sSum[N][m];
for( i=0; i<N; i++){
    for( j=0; j<m; j++){
        sSum[i][j] = arr[i][j];
    }
}

for( i=0; i<N; i++){
    for( j=m-2 ; j>=0; j--){
        sSum[i][j]  +=  sSum[i][j+1];
    }
}
```

sSum row-wise

```
for( j=0; j<m; j++){
    for( i=N-2; i>=0; i--){
        sSum[i][j]  +=  sSum[i+1][j];
    }
}
```

sSum column-wise

```
long ans = sSum[N-1][m-1];      //ans = INT_MIN
    for( i=0; i<N; i++){
        for( j=0; j<m; j++){
            ans = Max( ans, sSum[i][j]);
        }
    }
return ans;
```

T.C → O(N*m)
S.C → O(N*m)

Q: Given a matrix [N][M] where every row and column are sorted. find an element K.

	0	1	2	3	4	5
0	-1	2	4	5	9	11
1	1	4	7	8	10	14
2	3	7	9	10	12	18
3	6	10	12	14	16	20
4	9	13	16	19	22	24
5	11	15	19	21	24	27
6	14	20	25	29	31	39
7	18	24	29	32	34	42

todo

K = 25.

idea-1.
iterate on each and every element.
T.C → $O(N*M)$

idea-2.
B.S on every row.
T.C → $O(N \cdot \log M)$

code →

```
i = 0,  j = M-1
while( i < N && j >= 0){
    if (arr[i][j] == K){
        return true;
    }
    else if (arr[i][j] < K){
        i++;
    }
    else {
        j--;
    }
}
return false;
```

T.C → $O(N+M)$
S.C → $O(1)$

——————x——————x——————

→ psum[][]

→ sSum[][]

→ similar to B.S

learning phase.

↓

implement that question by yourself. ⇒ Revision