Today's Content

→ Submatrix Sum Queries
→ Sum of all submatrices
→ Max Submatrix Sum

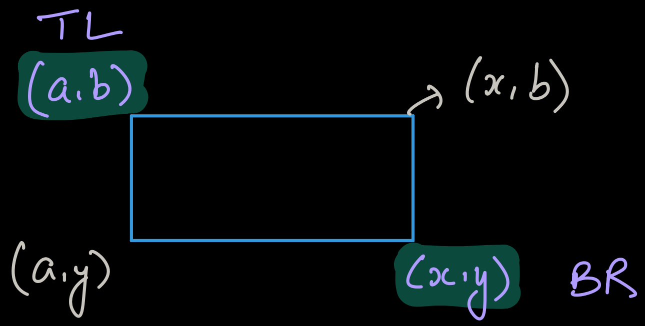
Qn: Given a matrix of size N * M, for each query q, find the sum of given submatrix.

↳ part of the matrix (continuous part of matrix)

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

start row, start col.
end row, end col.

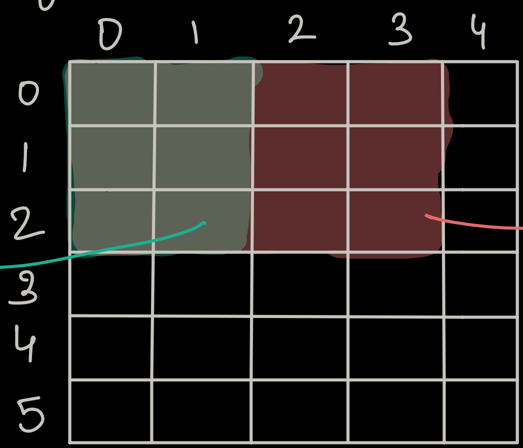


Q

TL	BR	
(2,1)	(4,2)	= 26
(0,0)	(1,1)	= 6

idea: For every query, traverse the submatrix & get the sum.
TC: O(Q * N * M)
SC: O(1)

idea 2: Use pfsum[]

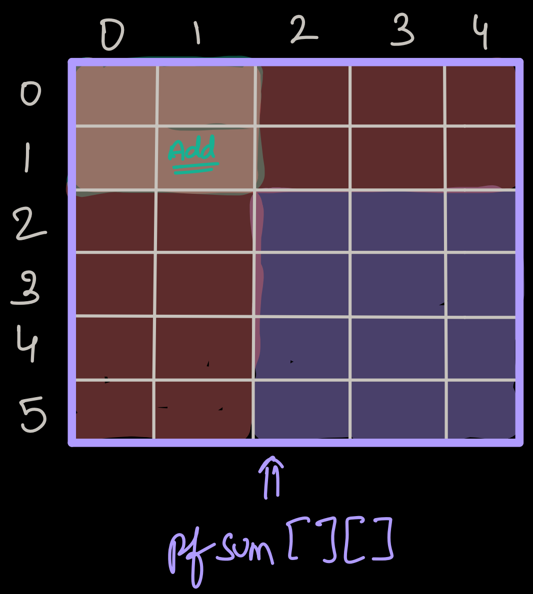


pfsum[i][j] will contain the sum from (0,0) to (i,j).

Sum from (0,0) to (2,1)

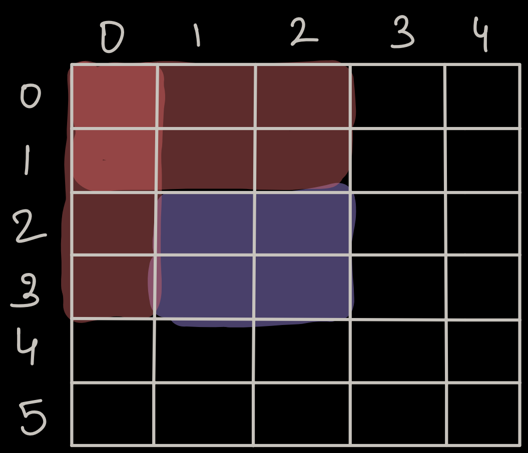
Sum from (0,0) to (2,3)
pf[2][3]

* Assume the mat below is pfsum[][].



<u>sum (2,2) to (5,4)</u>

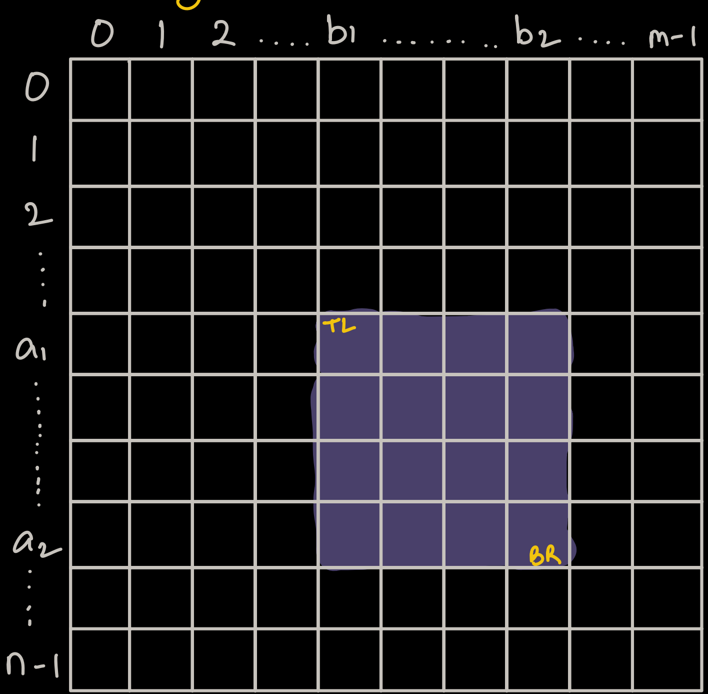
pf[5][4] − pf[1][4] − pf[5][1] + pf[1][1]



sum(2,1) to (3,2)

pf[3][2] − pf[1][2] − pf[3][0] + pf[1][0].

// Generalize it



TL (a_1, b_1) BR (a_2, b_2)

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

TC: $O(N*M) + O(Q*1)$

SC: $O(N*M)$

Qn: How to create pf array?

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

$\Downarrow$

apply pf sum on every row $\rightarrow$

	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	$a_2+b_2+c_2$

finding row-wise sum

TC: $O(N*M)$

$\Downarrow$

apply pf sum on every col $\downarrow$

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

finding col-wise sum.

TC: $O(N*M)$

Overall: $O(N*M)$

Sum (1,1) to (2,2)

$pf[2][2] - pf[0][2] - pf[2][0] + pf[0][0]$

$$\begin{array}{l}\cancel{a_0}+\cancel{b_0}+\cancel{c_0}\\ \cancel{a_1}+b_1+c_1\\ \cancel{a_2}+b_2+c_2\end{array} - (\cancel{a_0}+\cancel{b_0}+\cancel{c_0}) - \left(\begin{array}{c}a_0+\cancel{a_1}+\\ \cancel{a_2}\end{array}\right) + a_0$$

$\boxed{\begin{array}{l}b_1+c_1\\ b_2+c_2\end{array}}$ $\quad -\cancel{a_0} + \cancel{a_0}$

Pseudocode :-

```
void submatrix sum( int [][] mat) {
    int pf [N][M]
    // Row Sum
    for (i= 0; i < N; i++) { // for every row
        pf[i][0] = mat[i][0]
        for (j=1; j<M; j++) {
            pf[i][j] = pf [i][j-1] + mat[i][j]
        }
    }

    // Col Sum
    for ( j = 0; j<M; j ++) { // for every col.
        for(i=1; i < N; i++) {
            pf [i][j] = pf [i-1] [j] + pf [i][j]
        }
    }

    while (Q-- > 0) {
        // Given (a1 , b1)  (a2 , b2)
        Sum = pf [a2][b2]

        if (a1-1 >= 0) { sum = sum - pf [a1-1][b2] }
        if ( b1-1 >= 0) { sum = sum - pf [a2] [b1-1] }
        if (a1-1>=0 && b1-1 >=0) { sum = sum + pf [a1-1] [b1-1] }

        print (sum)
    }
}
```

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

	0	1	2
0	1	1	1
1	2	2	2
2	3	3	3

$\Rightarrow$ row sum

1	2	3
2	4	6
3	6	9

$\Rightarrow$ col sum

	0	1	2
0	1	2	3
1	3	6	9
2	6	12	18

pf [ ][ ]

$pf[2][2] - pf[0][2] - pf[2][0] + pf[0][0]$

$18 - 3 - 6 + 1 = 10$

<u>Qn</u> : Given a matrix of size $N \times M$. Calculate sum of all submatrices.

<u>Eg</u>:-

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

<u>Size 1:</u>

$[3]$ $[1]$

$[-1]$ $[-2]$

$[2]$ $[4]$

<u>Size 2:</u>

$[3 \;\; 1]$

$[-1 \;\; -2]$

$[2 \;\; 4]$

$\begin{bmatrix} 3 \\ -1 \end{bmatrix}$ $\begin{bmatrix} 1 \\ -2 \end{bmatrix}$

$\begin{bmatrix} -1 \\ 2 \end{bmatrix}$ $\begin{bmatrix} -2 \\ 4 \end{bmatrix}$

<u>Size 3</u>

$\begin{bmatrix} 3 \\ -1 \\ 2 \end{bmatrix}$ $\begin{bmatrix} 1 \\ -2 \\ 4 \end{bmatrix}$

<u>Size 4</u>

$\begin{bmatrix} 3 & 1 \\ -1 & -2 \end{bmatrix}$

$\begin{bmatrix} -1 & -2 \\ 2 & 4 \end{bmatrix}$

<u>Size 6</u>

$\begin{bmatrix} 3 & 1 \\ -1 & -2 \\ 2 & 4 \end{bmatrix}$

<u>ele</u>	<u>freq</u>	
3	6	$\longrightarrow 3 \times 6 = 18$
1	6	$\longrightarrow 1 \times 6 = 6$
-1	8	$\longrightarrow -1 \times 8 = -8$
-2	8	$\longrightarrow -2 \times 8 = -16$
2	6	$\longrightarrow 2 \times 6 = 12$
4	6	$\longrightarrow 4 \times 6 = 24$
		36

Contribution Technique.

<u>idea</u>: Use contribution technique & figure out how many submatrix is the element present in.

	0	1	2	3	4
0	T	T	T		
1	T	T	T		
2	T	T	TB	B	B
3			B	B	B
4			B	B	B
5			B	B	B

Qn: In how many submatrix is (2,2) present?

TL, BR

$\Rightarrow 9 \times 12 = 108$

	0	1	2	3	4
0	T	T	T		
1	T	T	T*B	B	B
2			B	B	B
3			B	B	B
4			B	B	B
5			B	B	B

Qn: In how many submatrix is (1,2) present?

$6 \times 15 = 90$

// Generalize

	0	1 …	j	…	m-1
0	T	T	T		
1	T	T	T		
⋮	T	T	T		
i	T	T	TB	B	B
⋮			B	B	B
n-1			B	B	B

Top left: $[0\ i] * [0\ j]$
$= (i+1)(j+1)$

Bottom Right: $[i\ \ n-1] * [j\ \ m-1]$
$= n-\cancel{1}-i+\cancel{1} \qquad m-\cancel{1}-j+\cancel{1}$
$= (n-i)*(m-j)$

No. of submatrix in which (i,j) will be present. $\Rightarrow (i+1)(j+1)(n-i)(m-j)$

```
sum=0
for (i=0; i<n; i++) {
    for(j=0; j<m; j++) {
        sum = sum + (i+1)(j+1)(n-i)(m-j)*mat[i][j]
    }
}
```

TC: O(N*M)

Qn: Given a matrix N×M. Find max submatrix sum.

Note: submatrix starts at row = 0
ends at row = n−1

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

squeezed it into 1D array.

Sum of every row: −2 4 −1 5 −3 1 ⇒ apply Kadane's algo

Qn: Given a mat [N][M]. Find the max submatrix sum where submatrix starts from row = 0. end at any row.

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

Start = 0, end = 0

2 −4 1 3 −1 2 = 5

start = 0, end = 1

3 −1 3 −4 2 5 = 8

Start = 0, end = 2

3 −2 4 −1 6 −2 = 10

start = 0, end = 3

4 −3 −2 3 2 4 = 9

Pseudo code :

```
ans = -∞
sum[m] = {0}   // Create a sum array of size = m & initialize to 0
for(e = 0; e < n; e++){
    for(j = 0; j < m; j++){
        sum[j] += mat[e][j]
    }
    ans = max(ans, Kadane(sum, m))
}
return ans
```

Qn: Given a mat[N][M]. Find the max submatrix sum where submat. start = any row
end = any row

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

Pseudo code :

```
ans = -∞

for(start = 0; start < n; start++){
    sum[m] = {0}   // Create a sum array of size = m
    for(e = start; e < n; e++){
        for(j = 0; j < m; j++){
            sum[j] += mat[e][j]
        }
        ans = max(ans, Kadane(sum, m))
    }
}
return ans
```

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