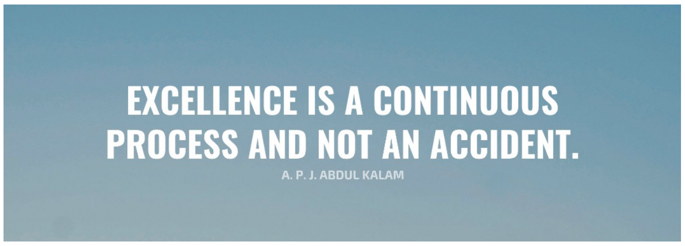


Todays Content

- → Sub-matrix sum queries
- → Sum of all sub-matrices
- → max submatrix sum.

// for every query, find sub-array sum.

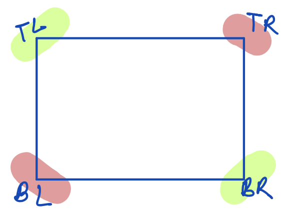
pSum[i] → Sum of elements from [0 - i]

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

- → part of matrix.
- → whole matrix / single 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 & BR should be fixed
or TR & BL should be fixed
then sub-matrix is defined]

For our entire discussion: TL & BR.

Query:

① TL (2,1) BR (4,2)

ans → 9 + 8 + (-1) + 2 + 2 + 6
= 26.

② TL (1,1) BR (3,3)

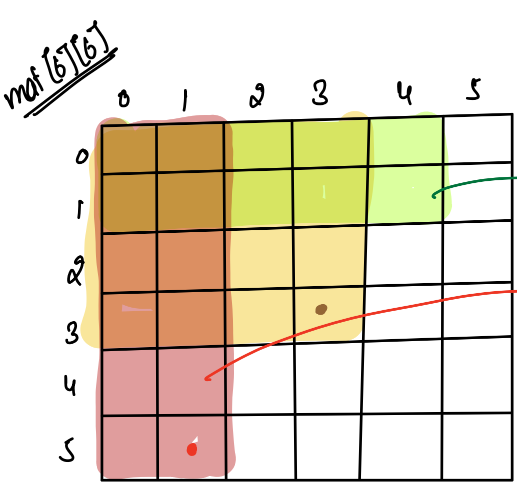
ans → 33.

idea 1: For every query, iterate over submatrix and calculate sum.

T.C → $O(QNM)$, S.C → $O(1)$

idea 2: pf [ ] [ ] : prefix matrix.

pf [i][j] → Sum of all elements from [0,0] to [i,j]

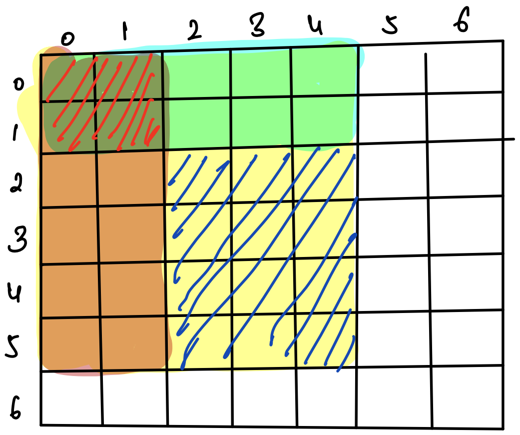


pf [1][4] = sum [0,0] to [1,4]

pf [5] [1] = sum [0,0] to [5,1].

pf [3][3] = sum [0,0] to [3,3]

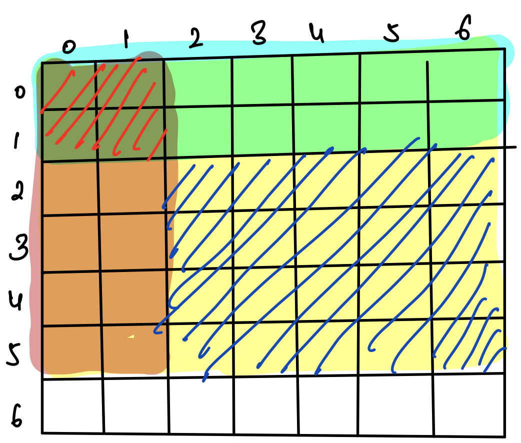
//Assume pf[][] ✓



Sum[TL 2,2 BR 5,4]

pf[5][4] - pf[5][1] - pf[1][4] + pf[1][1].

pf[][] ✓

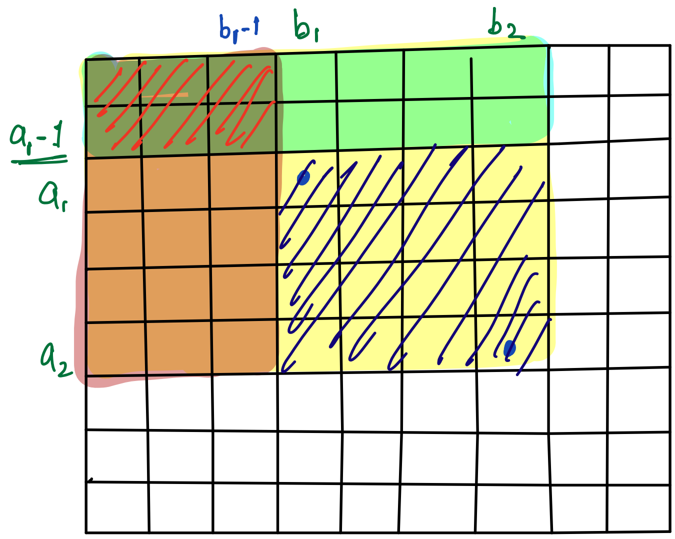


Query →

TL (2,2) BR (5,6)

pf[5][6] - pf[5][1] - pf[1][6] + pf[1][1].

// Generalisation $pf[\,][\,]$ ✓



$\underline{TL}$ $\underline{BR}$

Q: (a_1, b_1) (a_2, b_2)

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

Generalisation. $\underline{TL}$ & $\underline{BR}$.
(a_1, b_1) (a_2, b_2)

if $(b_1 > 0)$ if$(a_1 > 0)$ if $(a_1 > 0$ && $b_1 > 0)$

$$\left[\; ans. = pf[a_2][b_2] - pf[a_2][b_1-1] - pf[a_1-1][b_2] + pf[a_1-1][b_1-1] \;\right]$$

*
$$\boxed{\begin{array}{l} T.C \rightarrow O(Q + N.M) \\ S.C \rightarrow O(1) \end{array}}$$

$\leftrightarrow$ if you modify the given matrix.

$O(N*M) \rightarrow$ if you are taking $pf[\,][\,]$ matrix

Calculating pf[ ][ ]

mat[ ][ ] → Given matrix

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

on every row, apply pSum.

	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	$a_2+b_2+c_2$

$$pf[1][2] = a_0+b_0+c_0+a_1+b_1+c_1$$

$$pf[2][1] = a_0+b_0+a_1+b_1+a_2+b_2$$

on every col, apply pSum.

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

TC → pSum on every row + pSum on every column.

$$\Downarrow \qquad\qquad \Downarrow$$

$$(N * M) + (M * N) = 2NM$$

$$\boxed{T.C \rightarrow O(N*M)}$$

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

Eg:

$$\begin{array}{c|cc} & 0 & 1 \\ \hline 0 & 3 & 1 \\ 1 & -1 & -2 \\ 2 & 2 & 4 \end{array} \quad \begin{bmatrix} 3 & 1 \\ -1 & -2 \\ 2 & 4 \end{bmatrix} = \left\{ \begin{array}{l} [3] \quad [3,1] \quad \begin{bmatrix} 3 & 1 \\ -1 & -2 \end{bmatrix} \quad \begin{bmatrix} 3 \\ -1 \end{bmatrix} \quad \begin{bmatrix} 3 \\ -1 \\ 2 \end{bmatrix} \\ \begin{bmatrix} 3 & 1 \\ -1 & -2 \\ 2 & 4 \end{bmatrix} \quad [1] \quad \begin{bmatrix} 1 \\ -2 \end{bmatrix} \quad \begin{bmatrix} 1 \\ -2 \\ 4 \end{bmatrix} \quad [-1] \quad [-1 \ -2] \\ \begin{bmatrix} -1 \\ 2 \end{bmatrix} \quad \begin{bmatrix} -1 & -2 \\ 2 & 4 \end{bmatrix} \quad [-2] \quad \begin{bmatrix} -2 \\ 4 \end{bmatrix} \quad [2] \quad [2 \ 4] \\ [4] \end{array} \right\}$$

element	count
(3	* 6)
+ (1	* 6)
+ (-1	* 8)
+ (-2	* 8)
+ (2	* 6)
+ (4	* 6)

$$= 36$$

idea: Contribution technique.

For every element, we need to find, how many times that element is appearing in all the sub-matrices.

// How to fix sub-matrix → TL, BR.

idea :

	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

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

Note : To fix submatrix.

TL (a_1, b_1) ⟹ options for top-left = 12

BR (a_2, b_2) ⟹ options for B.R. = 9.

$a_2 >= a_1$
$b_2 >= b_1$

[ans. = 12×9 = 108.]

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

T.L	B.R.
0,0	1,2
0,1	1,3
0,2	1,4
1,0	2,2
1,1	2,3
	2,4
1,2	3,2
	3,3
	3,4

// In how many submatrices (1,2) is present?

To fix a submatrix.

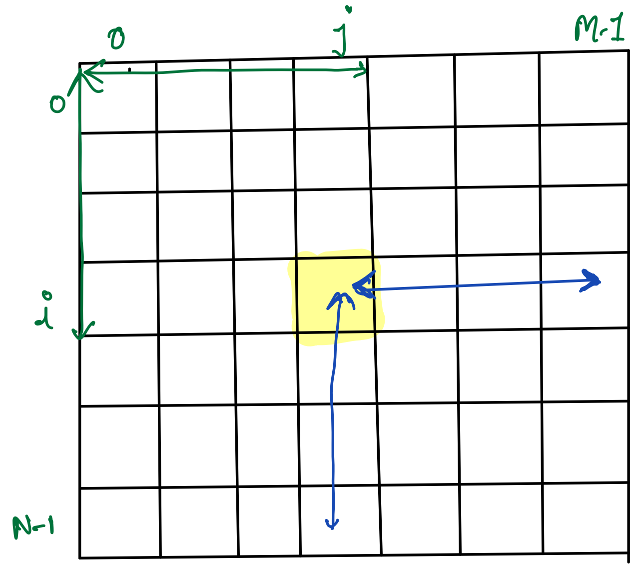
TL (a_1, b_1) → options for T.L. → 6

BR (a_2, b_2) → option for B.R → 9

6 * 9

ans = 54.

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



To fix a sub-matrix

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

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

$$ans = (i+1) * (j+1) * (N-i) * (M-j)$$

$[a, b]$

$\Rightarrow b-a+1$

$[i, N-1]$

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

pseudo-code:

```
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
```

$$\boxed{\begin{aligned} T.C &\rightarrow O(N*M) \\ S.C &\rightarrow O(1) \end{aligned}}$$

[Break 11:58 → 12:05]

Q) Given a matrix with dimensions N*M.

P1 → find max-submatrix sum,
where, submatrix starts from 0^{th} row & ends at $N-1^{th}$ row.
(last)

Eg:

$N = 3$
$M = 6$

	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

In any valid sub-matrix →
All elements of columns must be included.

	0	1	2	3	4	5
arr[] →	-2	4	-1	5	-3	1

ans = 8.

Q. Given a matrix. Find max sub-matrix sum.

Note → where submatrix starts at row = 0 & can end anywhere

end = [0, 1, 2, 3].

mat[4][6]

	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

arr →

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

ans.

start = 0 , end = 0

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

5

1	3	2	-7	3	3

start = 0 , end = 1

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

8

0	-1	1	3	4	-7

start = 0 , end = 2

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

10 ← final ans.

1	-1	-6	4	-4	6

start = 0, end = 3.

4	-3	-2	3	2	4

9.

Pseudo-code:

```
int[] a = new int[M];
for( st = 0; st < N; st++){
    for( end = st; end < N; end++){
        //Add elements of the curr row (end) in a[]
        for( i = 0; i < M; i++){
            a[i] += arr[end][i];
        }
        ans = Max(ans, Kadanes(a[], M));   O(M)
    }
}
return ans.
```

$T.C \rightarrow O(N^2M)$
$S.C \rightarrow O(M)$

Maximum sub-matrix sum, where row can start anywhere & it can end anywhere.

$$|x| \begin{cases} x, & x > 0 \\ -x, & x < 0 \end{cases}$$

P.S.S.

$$|a-b| \begin{cases} \underline{a-b}, & a > b \\ \underline{b-a}, & a < b \end{cases}$$

flip.

arr = [0 1 0 0 1 1 0 1 0]

L, R → flip all the characters.

Atmost → 1 **operation**.

[Maximise no. of 1's]

arr = [1 1 1 1 1 1]

$$\begin{array}{ccccccccccccc} & & & & 1 & 1 & 1 & 0 & 1 & 1 & & \\ \text{arr} = & [\,1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1\,] \\ & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \end{array}$$

arr → [1 0 0 1 0 0 1 1 1 0 1 0 1 0 0 1]

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

no. of zeros & no. of 1's.

	0's.	1's.
3 — 10	5	3

after flipping.

0's.	1's.
3	5.

Net gain of 1's = (2)

	no. of 0's.	no. of 1's.	Net gain of 1's.
sub1 →	8	3	5
sub2	7	1	6.

Which subarray are you going to flip?

Instead of looking for a subarray with maximum 0's, look for a subarray that is giving you maximum net gain of 1's.

	net gain of 1's.
0 → 1	[+1]
1 → 0	[-1]

[Problem Solving Session]