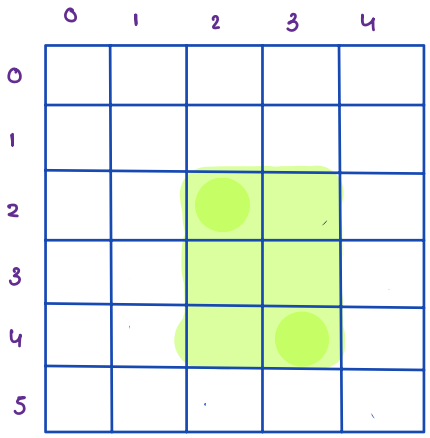
Agenda

1. Submatrix sum queries
2. Max submatrix sum
3. Sum of all submatrices

Representing a submatrix



TL → 2, 2

BR → 4, 3

Top-left and bottom right corners are required.

Q.1 Given a N×M matrix and queries, for each query find submatrix sum.

queries

(TL)		(BR)		
x1	y1	x2	y2	ans
2	1	4	3	20
3	2	5	4	36

A ⇒

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

i) brute force : go on every query, and travel that submatrix to find sum.

O(N × M × Q)

ii) Improvise : prefix sum

A ⇒

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

Prefix sum

	0	1	2	3	4
0	7			5	
1		23	16		
2					
3					
4					
5					

psf[i][j] → sum of matrix from 0,0 to i,j

(TL) (BR)

A :

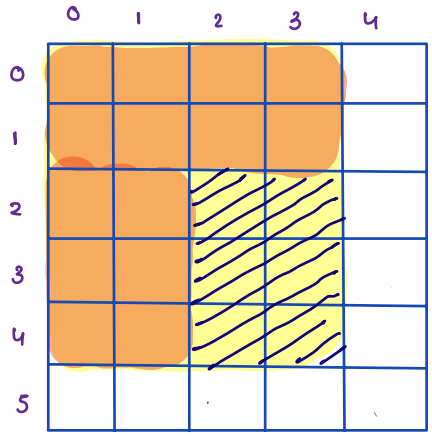
	0	1	2
0	a	b	c
1	d	e	f
2	g	h	i

apply row-wise prefix sum →

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

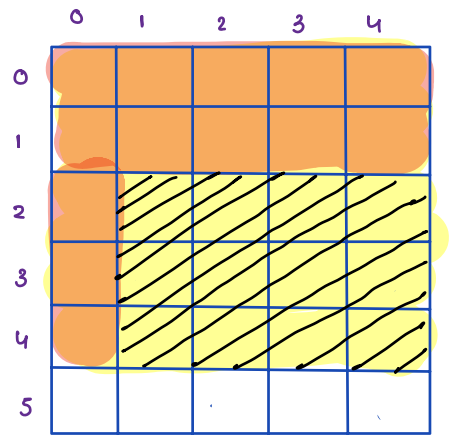
apply col-wise prefix sum ↓

	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



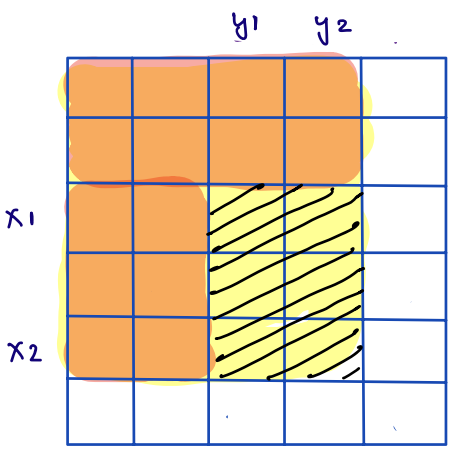
TL (2,2) BR (4,3)

$ps[4][3] - ps[1][3] - ps[4][1] + ps[1][1]$



TL (2,1) BR (4,4)

$ps[4][4] - ps[4][1] - ps[1][4] + ps[1][1]$

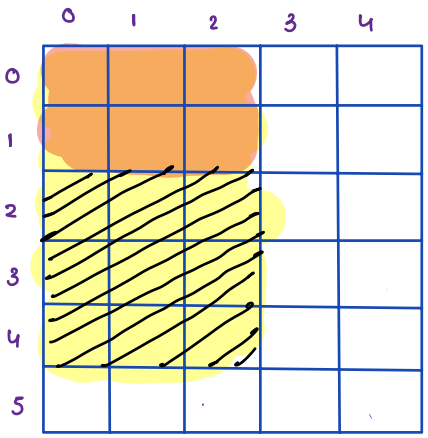


TL (x_1, y_1) BR (x_2, y_2)

$ps[x_2][y_2] - ps[x_1-1][y_2] - ps[x_2][y_1-1] + ps[x_1-1][y_1-1]$

$$ps[x2][y2] - ps[x1-1][y2] - (ps[x2][y1-1] - ps[x1-1][y1-1])$$

$$\Rightarrow ps[x2][y2] - ps[x1-1][y2] - ps[x2][y1-1] + ps[x1-1][y1-1]$$



TL (2,0) → $x_1\ y_1$

BR (4,2) → $x_2\ y_2$

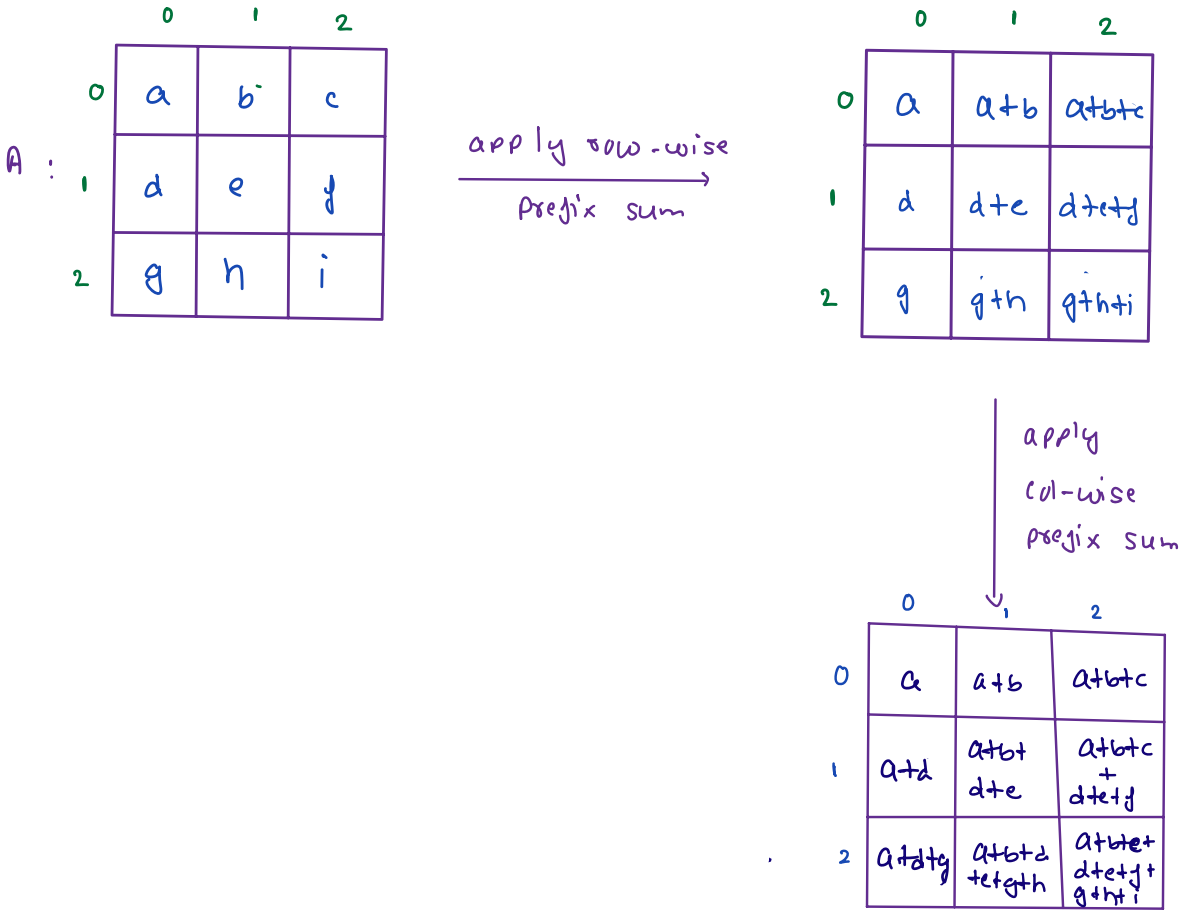
$$ps[4][2] - ps[1][2] - ps[4][-1] + ps[1][-1]$$

```
void submatrixSumQuery(int[][] A, int[][] Q){
    int[][] ps = prefixSum2D(A);

    for(int i=0; i<Q.length; i++){
        int x1 = Q[i][0];
        int y1 = Q[i][1];
        int x2 = Q[i][2];
        int y2 = Q[i][3];

        int sum = 0;

        sum += ps[x2][y2];

        if(x1>0)
        sum -= ps[x1-1][y2];

        if(y1>0)
        sum -= ps[x2][y1-1];

        if(x1>0 && y1>0)
        sum += ps[x1-1][y1-1];

        System.out.println(sum);
    }
}
```

itr: $2(N*M)+Q$

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

SC: $O(N*M)$


A :
0 1 2
0 a b c
1 d e f
2 g h i
apply row-wise
prefix sum
0 1 2
0 a a+b a+b+c
1 d d+e d+e+f
2 g g+h g+h+i
apply
col-wise
prefix sum
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+e+
d+e+g+
g+h+i

```
int [][] prefixSum2D(int [][] A){
    int n = A.length;
    int m = A[0].length;
    int [][] ps = new int [n][m];
    // apply row by row ps
    for(int i=0; i<n; i++){
        for(int j=0; j<m; j++){
            if(j==0){
                ps[i][j] = A[i][j];
            }
            else{
                ps[i][j] = ps[i][j-1] + A[i][j];
            }
        }
    }
    //apply col by col ps
    for(int j=0; j<m; j++){
        for(i=1; i<n; i++){
            ps[i][j] = ps[i-1][j] + ps[i][j];
        }
    }
    return ps;
}
```

Q.2 Given row wise and column wise sorted matrix, find maximum submatrix sum.

A =

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

max element is always a part of our ans and max element is present at (n-1, m-1)

In ans submatrix BR is fixed to (n-1, m-1)

⇓

Go on all TL possible and find the best ans.

A =

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

A =

	0	1	2
0	-50	-40	-30
1	-25	-20	-15
2	-19	-14	-3

TL

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

A =

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

```
int maxSumSubmatrix(int[][] A) {
    int[][] ps = prefixSum2D(A);
    int n = A.length;
    int m = A[0].length;
    int x2 = n-1;
    int y2 = m-1;                    } → BR
    int ans = Integer.MIN_VALUE;
    for(int i=0; i<n; i++) {
        for(int j=0; j<m; j++) {     } → go on all possible TL
            int x1 = i, y1 = j;
            // sum of submatrix  TL: x1, y1 & BR: x2, y2
            int sum = 0;
            sum += ps[x2][y2];
            if(x1 > 0)
                sum -= ps[x1-1][y2];
            if(y1 > 0)
                sum -= ps[x2][y1-1];
            if(x1 > 0 && y1 > 0)
                sum += ps[x1-1][y1-1];
            ans = Math.max(ans, sum);
        }
    }
    return ans;
}
```

Q-3 Given a N X M Submatrix, find sum of all submatrices sum.

	0	1
0	2	1
1	3	4

TL	BR	Sum
0,0	0,0	2
0,0	0,1	3
0,0	1,0	5
0,0	1,1	10
0,1	0,1	1
0,1	1,1	5
1,0	1,0	3
1,0	1,1	7
1,1	1,1	4
		+
		40

Improvised idea

if A[i][j] is coming in x no. of matrices then its contribution if final ans is: x * A[i][j]

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

A[i][j] is coming in how many submatrices.

valid TL → 12

valid BR → 9

12 * 9 = 108

			j		m-1
	0	1	2	3	4
0	TL	TL	TL		
1	TL	TL	TL		
2	TL	TL	TL		
i 3	TL	TL	TL BR	BR	BR
4			BR	BR	BR
n-1 5			BR	BR	BR

valid TL : (i+1) * (j+1)

valid BR : (n-i) * (m-j)

freq => (i+1) * (j+1) * (n-i) * (m-j)

```
int solve(int[][] A) {
    int n = A.length;
    int m = A[0].length;

    for(int i=0; i<n; i++) {
        for(int j=0; j<m; j++) {
            int freq = (i+1)*(j+1)*(n-i)*(m-j);
            ans += freq*A[i][j];
        }
    }

    return ans;
}
```

Doubts

$(i+1)*(j+1) * (n-i)*(m-j)$;

$n=2, m=2$

	0	1
0	2	1
1	3	4

(i, j)	freq	ans
0, 0	1 * 1 * 2 * 2 = 4	8
0, 1	1 * 2 * 2 * 1 = 4	4
1, 0	2 * 1 * 1 * 2 = 4	12
1, 1	2 * 2 * 1 * 1 = 4	16

Q. No. of minimum swaps to make all ele <= B together.

A: [1 10 15 2 9 13 4] B = 5
0 1 2 3 4 5 6 ans = 2

A: [25 30 2 18 7 6 9 50 33] B = 10
0 1 2 3 4 5 6 7 8 ans = 1

A: [19 11 3 9 7 25 6 20 4] B = 10
0 1 2 3 4 5 6 7 8 ans = 2

A: [19 11 3 9 7 25 6 20 4]

0	1	2	3	4	5	6	7	8

B = 10

k = 5 (<= B)

s	e	swaps
0	4	2
1	5	2
2	6	1
3	7	2
4	8	2

(no. ele > B , ans. of 1st window)

```
if (A[s-1] > B) {
    swaps--;
}
if (A[e] > B) {
    swaps++;
}
```