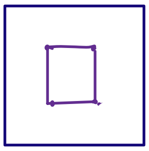
Agenda

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

Submatrix : Part of matrix is a submatrix.

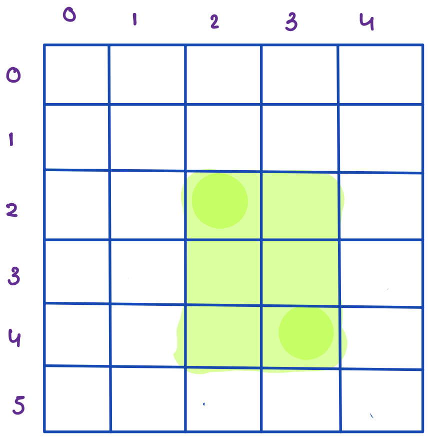


a) Single element is also a submatrix.

b) complete matrix is also a submatrix.

c) we just need opposite cornerns to get submatrix.

Top left & bottom right OR
Top right & bottom left



TL	BR
2,2	4,3

Q.1 Given a NXM 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
1	2	2	3	4

$$A \Rightarrow \begin{array}{c|ccccc} & 0 & 1 & 2 & 3 & 4 \\ \hline 0 & 7 & 1 & -6 & 3 & 12 \\ 1 & 10 & 5 & -1 & 0 & 9 \\ 2 & 6 & 4 & -3 & 8 & 11 \\ 3 & 13 & -8 & -5 & 12 & 4 \\ 4 & 3 & 2 & 1 & 9 & 8 \\ 5 & 4 & 3 & -2 & 6 & 3 \end{array}$$

i) Brute force : go on every query and then find sum that particular submatrix.

```
for ( go on every query ) {
        x1, y1        x2, y2
        int sum=0;
        for ( i= x1 ; i <= x2 ; i++ ) {
                                                    N*M
              for (j=y1 ; j<=y2 ; j++) {
                    sum += A[i][j];
              }
        }
        SOPln(sum);
}
```

TC: O(N*M*Q)

ii) optimised → prefix Sum

1D array:

ps[i] ⇒ sum of elements from 0 to i.

2D array:

ps[i][j] ⇒ sum of submatrix from 0,0 to i,j

TL BR

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

A

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

PS

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

A

apply row-wise → prefix sum

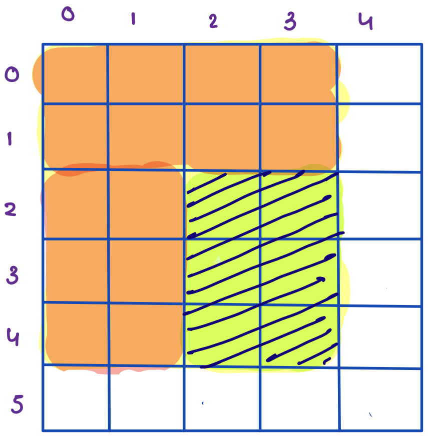
	0	1	2	3
0	3	5	9	10
1	-1	3	6	8
2	2	9	15	18

PS

↓ apply col-wise prefix sum

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

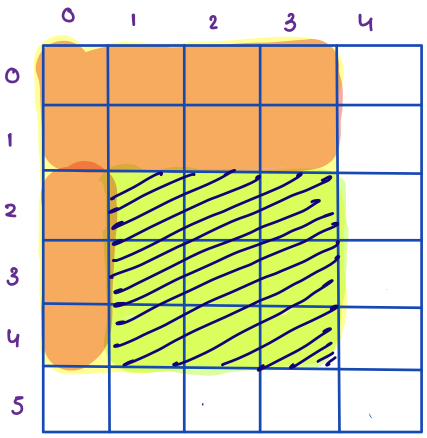
```
int [][]   prefixSum2D ( int [] [] A ) {
    int n= A.length;
    int m= A[0].length;
    int[][] ps = new int [n] [m];
    // apply prefix sum row by row
    for(int i=0; i<n; i++) {
        for (j=0; j<m; j++) {
            if ( j==0) {
                ps[i][j] = A[i][j];
            }
       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

    // apply prefix sum col by col
    for ( int j= 0; j<m; j++) {
        for ( int i=1 ; i<n; i++) {
            ps[i][j]  =  ps[i-1][j] + ps[i][j];
        }
    }
    return ps;
}
```



TL	BR
2,2	4,3

ps [4] [3] - ps[1] [3] -

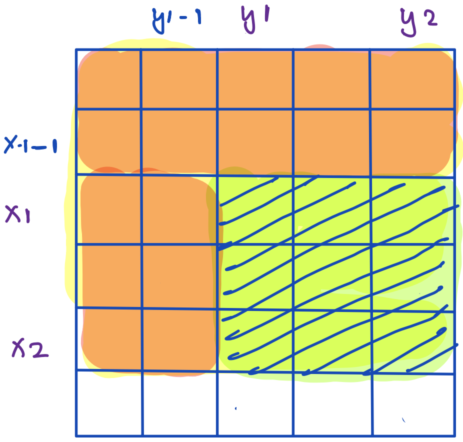
(ps[4] [1] - ps[1] [1])



TL	BR
2,1	4,3

ps [4] [3] - ps [1] [3] -

(ps [4] [0] - ps [1] [0])



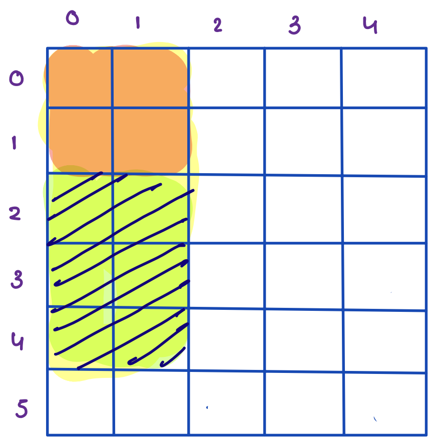
TL	BR
x1, y1	x2, y2

ps[x2] [y2] - ps[x1-1] [y2] -

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

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

ps [x2] [y2] - ps [x1-1] [y2] - ps [x2] [y1-1] + ps [x1-1] [y1-1]



TL	BR
(2,0)	(4,1)
x1, y1	x2, y2

PS[4][1] - PS[1][1] - ~~PS[4][-1]~~

+ ~~PS[1][-1]~~

ps[x2][y2] - ps[x1-1][y2] - ps[x2][y1-1] + ps[x1-1][y1-1]

$$
\begin{array}{c|ccccc}
 & 0 & 1 & 2 & 3 & 4 \\
\hline
0 & 3 & 2 & 4 & 1 & 6 \\
1 & -1 & 4 & 3 & 2 & 4 \\
2 & 2 & 7 & 6 & 3 & 2 \\
3 & 1 & 2 & 7 & 8 & 1
\end{array}
\xrightarrow{\text{Pymat[ ][ ]}}
\begin{array}{c|ccccc}
 & 0 & 1 & 2 & 3 & 4 \\
\hline
0 & 3 & 5 & 9 & 10 & 16 \\
1 & 2 & 8 & 15 & 18 & 28 \\
2 & 4 & 17 & 30 & 36 & 48 \\
3 & 5 & 20 & 40 & 54 & 67
\end{array}
$$

A

PS

Query

x1	y1	x2	y2
1	2	2	4

PS[2][4] - PS[0][4] - PS[2][1] + PS[0][1]

48 - 16 - 17 + 5 = 20

```
void   solve (int [ ] [ ]A, int [ ] [ ] Q ) {
      int [ ] [ ] PS = prefixSum2D (A);      } → 2*( N*M)

      for (int i=0; i< Q.length; i++)  {   → Q
            int x1 = Q[i][0];
            int y1 = Q[i][1];
            int x2 = Q [i] [2];
            int y2 = Q [i] [3];
            // find sum of submatrix  TL: x1, y1 & BR: x2,y2
            int sum= 0;
            sum+= PS[x2][y2];                     itr: 2(N*M)+Q

            if (x1 > 0)                           TC: O(N*M + Q)
            sum- = PS [x1-1] [y2];                SC:  O(N*M)

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

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

            SOPln(sum);
      }

}
```

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

$$A = \begin{array}{c|cccc} & 0 & 1 & 2 & 3 \\ \hline 0 & -20 & -16 & -4 & 8 \\ 1 & -10 & -8 & 2 & 14 \\ 2 & -1 & 6 & 21 & 30 \\ 3 & 5 & 7 & 28 & 42 \end{array}$$

$$A = \begin{array}{c|cccc} & 0 & 1 & 2 & 3 \\ \hline 0 & -20 & -16 & -4 & -1 \\ 1 & -10 & -8 & -2 & 5 \\ 2 & -4 & 2 & 4 & 8 \end{array}$$

$$A = \begin{array}{c|ccc} & 0 & 1 & 2 \\ \hline 0 & -50 & -40 & -30 \\ 1 & -25 & -20 & -15 \\ 2 & -19 & -14 & -3 \end{array}$$

$$A = \begin{array}{c|cccc} & 0 & 1 & 2 & 3 \\ \hline 0 & -20 & -16 & -4 & -1 \\ 1 & -10 & -8 & -2 & 5 \\ 2 & -4 & 2 & 4 & 8 \end{array}$$

In ans submatrix, we need to include the max. element.

↓

the max element of A[][] is always present n-1, m-1

↓

BR is already fixed n-1, m-1

go on all valid TL and find out best answer possible.

Top lefts

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

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

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

Q-3 Given a NXM matrix , find sum of all submatrices sum.

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

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

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

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

ans= 36

$$ans = 6*3 + 6*1 + 8*(-1) + 8*(-2) + 6*2 + 6*4$$

$$18 + 6 + -8 - 16 + 12 + 24 = 36$$

Conclusion: if A[i][j] is coming x submatrices so its contribution in ans will be

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

valid TL → 12

valid BR → 9

total submatrices = 12 * 9

= 108

			j		m-1
	TL	TL	TL		
	TL	TL	TL		
	TL	TL	TL		
i	TL	TL	TL BR	BR	BR
			BR	BR	BR
n-1			BR	BR	BR

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

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

total submatrices in which A[i][j] is present

$\Rightarrow (i+1)*(j+1) * (n-i)*(m-j)$

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

$n = 3$

$m = 2$

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

```
int ans= 0;

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];
    }
}
```

i j	freq	impact
0 0	1*1*3*2=6	18
0 1	1*2*3*1=6	6
1,0	2*1*2*2=8	-8
1,1	2*2*2*1=8	-16
2,0	3*1*1*2=6	12
2,1	3*2*1*1=6	24
		36

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

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

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

$6*3 + 6*1 + 8*(-1) + 8*(-2) + 6*2 + 6*4$