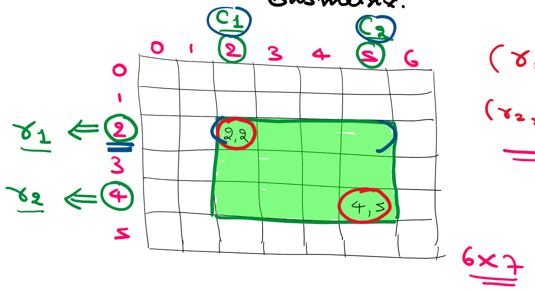
Arrays - 2

24 May 2023 21:02

Q Given a 2D matrix of size N×M.
Q queries ⇒ find the sum of a given submatrix.



$(r_1, c_1) \Rightarrow$ Top Left
$(r_2, c_2) \Rightarrow$ Bottom right

$r_2 \geq r_1$
$c_2 \geq c_1$

Sol^n

1) Brute Force

I/P ⇒ 1) N, M, Mat[N][M], Q
2) 2D Array of length Q×4 (Query)

i^{th} row ⇒ Query[i][0] = r_1
Query[i][1] = c_1
Query[i][2] = r_2
Query[i][3] = c_2

Code

```
for (i=0; i<Q; i++) {   // O(Q)
    // r1, c1, r2, c2
    sum = 0;
    for (j = r1; j <= r2; j++) {   //row iterator
        for (k = c1; k <= c2; k++) {   //col
            sum += Mat[j][k];
        }
    }
    print (sum);
}
```

O(N×M)

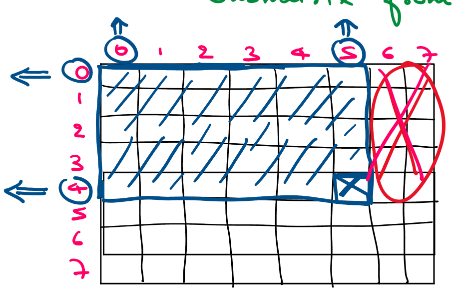
T.C. = O(Q×N×M)

2) Optimise ??

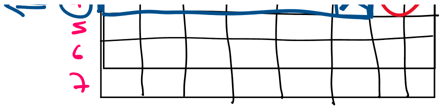
1D Array ⇒ Subarray Sum ⇒ Prefix Sum Array

2D Matrix ⇒ Submatrix Sum ⇒ Prefix Sum Matrix.

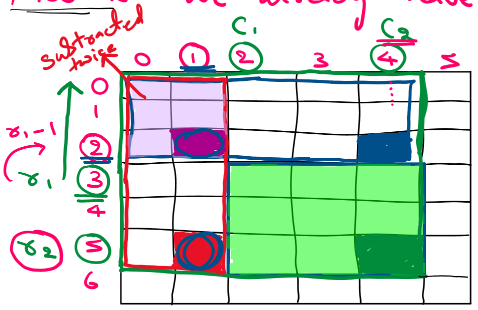
PS[i][j] ⇒ Sum of all elements in the submatrix from (0,0) to (i,j)

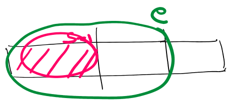


PS[4][5]



Assume : We already have a PS matrix.





PS[5][4] − PS[2][4] − PS[5][1] + PS[2][1]

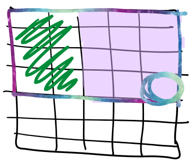
Sum [(r_1, c_1) to (r_2, c_2)] =
PS[r_2][c_2] − PS[r_1−1][c_2] − PS[r_2][c_1−1]
+ PS[r_1−1][c_1−1]

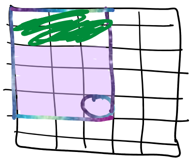
Edge Cases

1) If (r_1 == 0 && c_1 == 0) ⇒ PS[r_2][c_2]

2) If (r_1 == 0) ⇒ PS[r_2][c_2] − PS[r_2][c_1−1]

3) If (c_1 == 0) ⇒ PS[r_2][c_2] − PS[r_1−1][c_2]





	0	1	2
0	3	4	1
1	6	2	9
2	5	3	1

Mat

	0	1	2
0	3	7	8
1	9	15	25
2	14	23	34

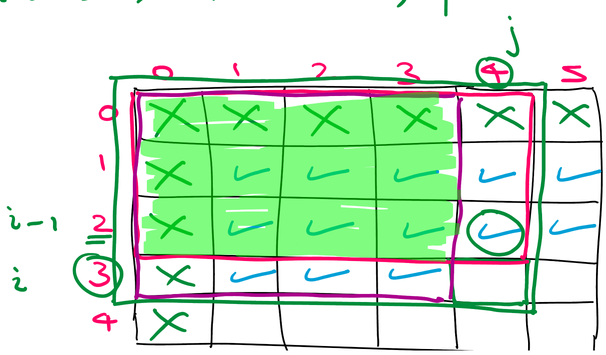
PS

PS[0][0] ⇒ (0,0) to (0,0)
PS[0][1] ⇒ (0,0) to (0,1)
PS[0][2] ⇒ (0,0) to (0,2)

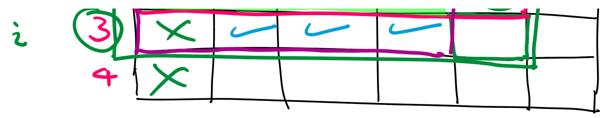
PS[1][0] ⇒ (0,0) to (1,0)
PS[1][1] ⇒ (0,0) to (1,1)

PS[1][2] = (0,0) to (1,2)

PS[2][0] ⇒ (0,0) (2,0)
PS[2][1] ⇒ (0,0) to (2,1)
PS[2][2] ⇒ (0,0) to (2,2)



PS[3][4]



Code

$$PS[i][j] = PS[i-1][j] + PS[i][j-1] - PS[i-1][j-1] + Mat[i][j]$$

```
PS[0][0] = M[0][0]
for (j=1; j<M; j++) {   // 1st row
    PS[0][j] = PS[0][j-1] + Mat[0][j];
}
for (i=1; i<N; i++) {   // 1st column.
    PS[i][0] = PS[i-1][0] + Mat[i][0];
}
for (i=1; i<N; i++) {
    for (j=1; j<M; j++) {
        PS[i][j] = PS[i-1][j] + PS[i][j-1]
                   - PS[i-1][j-1] + Mat[i][j];
    }
}
```

$T.C. = O(N \times M)$

Overall $TC = O(N \times M) + O(Q \times 1)$

$S.C. = O(N \times M)$

Q-2 Given a matrix of size $N \times M$. Calculate the sum of all submatrix sums

$$M = \begin{array}{c|c} & \begin{array}{ccc} 0 & 1 & 2 \end{array} \\ \begin{array}{c} 0 \\ 1 \end{array} & \begin{bmatrix} 4, & 9, & 6 \\ 5, & -1, & 2 \end{bmatrix} \end{array}$$

$\Rightarrow$ $[4] = 4$, $[9] = 9$, $[6] = 6$, $[5] = 5$, $[-1] = -1$, $[2] = 2$

$[4, 9] = 13$, $[9, 6] = 15$, $[5, -1] = 4$, $[-1, 2] = 1$, $[4, 9, 6] = 19$, $[5, -1, 2] = 6$

$\begin{bmatrix} 4 \\ 5 \end{bmatrix} = 9$ $\quad \begin{bmatrix} 6 \\ 2 \end{bmatrix} = 8$

$\begin{bmatrix} 9 \\ -1 \end{bmatrix} = 8$

$\begin{bmatrix} 4, 9 \\ 5, -1 \end{bmatrix} = 17$ $\quad \begin{bmatrix} 9, & 6 \\ -1, & 2 \end{bmatrix} = 16$

$[-1, 2] = 16$

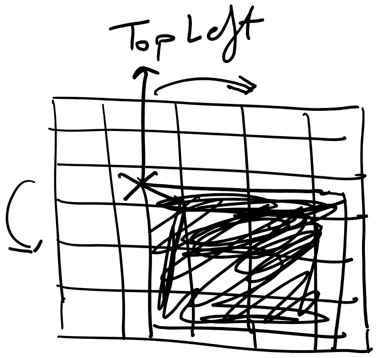
$$\begin{bmatrix} 4 & 9 & 6 \\ 5 & -1 & 2 \end{bmatrix} = 25 \qquad \text{Sum} = 166$$

$4\times 6 + 9\times 8 + 6\times 6 + 5\times 6 + (-1)\times 8 + 2\times 6.$

1) Brute force

⇒ for all submatrices, compute sum & add.

(r_1, c_1, r_2, c_2) $\quad r_1 \le r_2$, $c_1 \le c_2$



Code

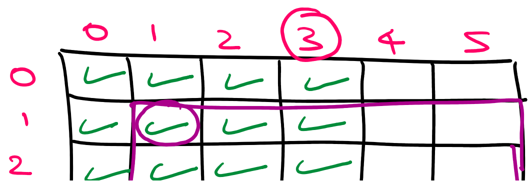
```
for (r1 = 0; r1 < N; r1++) {
    for (c1 = 0; c1 < M; c1++          } Top Left

        for (r2 = r1; r2 < N; r2++) {
            for (c2 = c1; c2 < M; c2++) {    } BR
                // (r1, c1) to (r2, c2)

                // Prefix Sum Matrix.
            }
        }
    }
}
```

T.C. $= O(N^2 M^2)$

S.C. $= O(N\times M)$

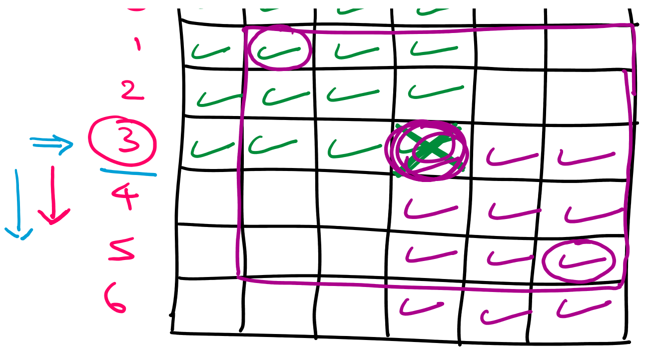
Array ⇒ Sum of all Subarray Sums $= \sum_{\forall ele} \left(\begin{matrix}\#\text{ Subarray an element is a part of}\end{matrix}\right) \times$ value

Matrix ⇒ Sum of all submatrix Sum $= \sum_{\forall \text{elements}} \left(\begin{matrix}\#\text{ submatrix an element is a part of}\end{matrix}\right) \times$ value



TL

BR

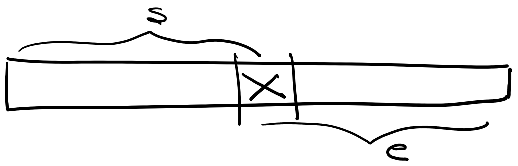


TL (r_1, c_1)

BR (r_2, c_2)

1) $r_1 \le i$
   $r_1 : [0, i]$
   $= (i+1)$

3) $r_2 \ge i$
   $r_2 : [i, N-1]$
   $= (N-i)$



2) $c_1 \le j$
   $c_1 : [0, j]$
   $= (j+1)$

4) $c_2 \ge j$
   $c_2 : [j, M-1]$
   $= (M-j)$

\# submatrices (i,j) is a part of $= (i+1)(j+1)(N-i)(M-j)$

Code

```
sum = 0;
for (i = 0; i < N; i++) {
    for (j = 0; j < M; j++) {
        count = (i+1)(j+1)(N-i)(M-j)
        contri = count x Mat[i][j];
        sum = sum + contri;
    }
}
```

T.C. $= O(N \times M)$, S.C. $= O(1)$

Doubt

A = [

$$A = [\,1\,0\,1\,1\,0\,1\,\boxed{0\,0\,1\,0\,1\,0\,0}\,1\,1\,1\,]$$

$$1\quad 1\quad 0\quad 1\quad 0\quad 1\quad 1$$

$$\begin{array}{r} +5 \\ -2 \\ \hline 3 \end{array}$$