Lecture ÷ Arrays 2

Agenda

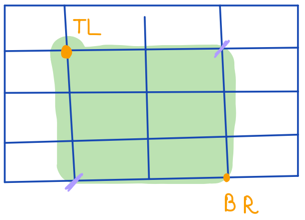
- Prefix sum [2D]
- Max submatrix sum
- Sum of all submatrix sums.

Qu1 Given mat[n][m].

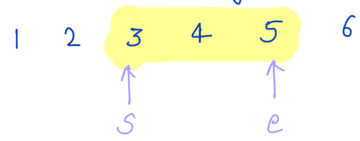
Q queries [top left & bottom right]

for every query, find submatrix sum.

submatrix



Subarray.



Eg:

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

Queries	sum
(1,1) (2,4)	16
⋮	⋮

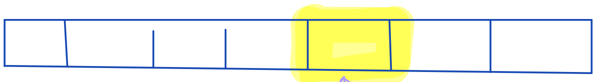
Brute force: Go to all query — O(Q)

for every query, sum of submatrix — O(n*m)

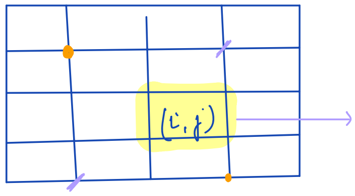
TC: O(Q * n * m).

challenge1 Prefix sum for 2D matrices

pf[] =



i [sum of all el from 0 to i]



sum[(0,0) to (i,j)]

	0	1	2
0	a0	b0	c0
1	a1	b1	c1
2	a2	b2	c2

prefix sum →

(0,0) to (0,1)

a0	a0 + b0	a0 + b0 + c0
a0 + a1	a0 + b0 + a1 + b1	a0 + b0 + c0 + a1 + b1 + c1
a0 + a1 + a2	a0 + b0 + a1 + b1 + a2 + b2	a0 + b0 + c0 + a1 + b1 + c1 + a2 + b2 + c2

	0	1	2
0	a0	b0	c0
1	a1	b1	c1
2	a2	b2	c2

prefix sum →

1. Take pf for each row.

2. Take pf for each col.

1	3	2
6	5	9
-2	7	8

pf →

1	4	6
~~6~~ 7	~~11~~ 15	~~20~~ 26
~~-2~~ 5	~~5~~ 20	~~13~~ 39.

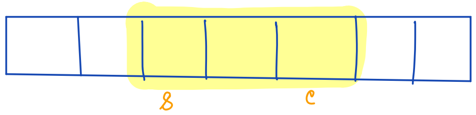
1 + 3 + 6 + 5

- 2 + 7

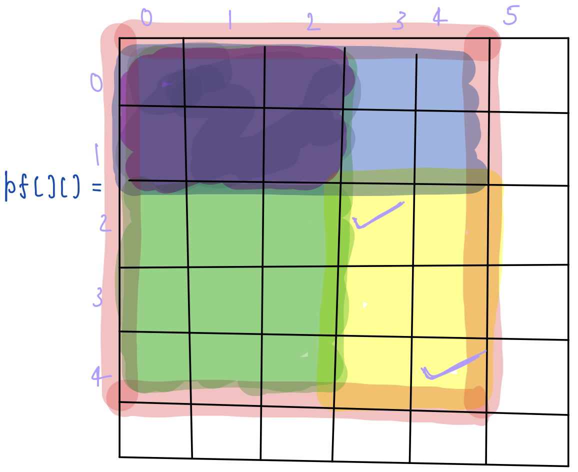
Challenge 2

sum $(a1, b1)$ to $(a2, b2)$

pf[] =



$sum = pf[e] - pf[s-1]$



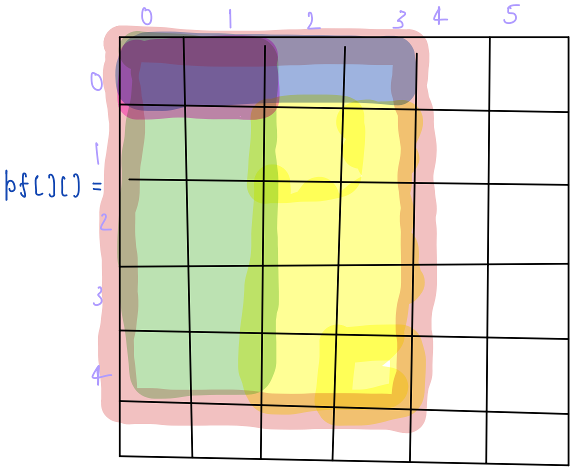
1. $sum[(\overset{a1}{2}, \overset{b1}{3}) \text{ to } (\overset{a2}{4}, \overset{b2}{4})] =$

$pf[\underset{a2}{4}][\underset{b2}{4}] - blue[pf[\underset{a1-1}{1}][\underset{b2}{4}]]$

$- green[pf[\underset{a2}{4}][\underset{b1-1}{2}]] +$

$common[pf[\underset{a1-1}{1}][\underset{b1-1}{2}]]$

2. $sum[(\underset{a1}{1}, \underset{b1}{2}) \text{ to } (\underset{a2}{4}, \underset{b2}{3})] =$



$pf[\overset{a2}{4}][\overset{b2}{3}] - pf[\overset{a1-1}{0}][\overset{b2}{3}]$

$- pf[\underset{a2}{4}][\underset{b1-1}{1}] + pf[\underset{a1-1}{0}][\underset{b1-1}{1}]$

Generalisation —

sum[(a1, b1) to (a2, b2)] = pf[a2][b2] —
pf[a1−1][b2] —
pf[a2][b1−1] +
pf[a1−1][b1−1]

Edge cases:

1) a1 = 0 [only]

sum[(a1, b1) to (a2, b2)] = pf[a2][b2] —
pf[a1−1][b2] —
pf[a2][b1−1] +
pf[a1−1][b1−1]

idx out— of bound

⇒ pf[a2][b2] —
pf[a2][b1−1]

2) b1 = 0 [only]

sum[(a1, b1) to (a2, b2)] = pf[a2][b2] —
pf[a1−1][b2] —
pf[a2][b1−1] +
pf[a1−1][b1−1]

3) a1==0 && b1==0

sum[(a1, b1) to (a2, b2)] = pf[a2][b2] —
pf[a1−1][b2] —
pf[a2][b1−1] +
pf[a1−1][b1−1]

Pseudocode:

1. calculate prefix sum.

```
int[][]  getPrefixsum(int[][] mat) {

        int  n = mat.length;

        int  m = mat[0].length;

        int [] [] pf = new int[n][m];

        // prefix on rows.

        for ( i=0;  i<n;  i++) {     — o(n*m)

              pf[i][0] = mat[i][0];

              for(j=1; j<m; j++) {

                  pf[i][j]  =  pf[i][j-1]  +  mat[i][j];

              }
        }


      // prefix on cols.

        for(j=0;  j<m;  j++) {   — o(n*m)

              for(i=1;  i<n;  i++) {

                    pf[i][j]  =  pf[i][j]  +  pf[i-1][j];
              }

        }
     return  pf;
}
```

2. for every query. calculate sum.

```
void  print subMatrixsum( int[][] mat,  int[][] queries) {

SC o(n*m) ——  int[][]  pf  =  getPrefixsum(mat);  — TC: [o(n*m)]

        for (i=0; i< queries.length; i++) {   — o(Q)

            int  a1  =  queries[i][0]
                 b1  =         [i][1]
                 a2  =         [i][2]
                 b2  =         [i][3]

        // formula  derived  above

        }
}
```

TC: o(Q + n*m)

SC: o(n*m)

Break: 8:32 AM

Qu2 Given mat[n][m], find sum of all submatrix sums

[Amazon, Google, Coinbase]

1	3
2	8

⇒

a) $1 = 1$

b) $1 \quad 3 = 4$

c) $\begin{matrix} 1 & 3 \\ 2 & 8 \end{matrix} = 14$

d) $3 = 3$

e) $2 = 2$

f) $8 = 8$

g) $\begin{matrix} 3 \\ 8 \end{matrix} = 11$

h) $\begin{matrix} 1 \\ 2 \end{matrix} = 3$

i) $2 \quad 8 = 10$

ans = 1 + 4 + 14 + 3 + 2 ---- 10

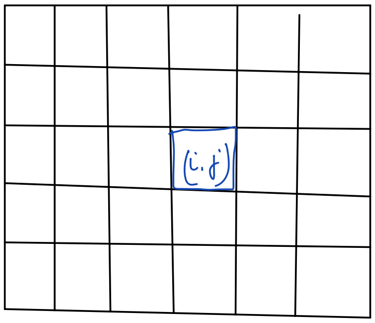
Brute force: Go to all submatrices — $O(n^2 m^2) \simeq O(n^4)$

```
TL  for( a1 = 0;  a1 < n;  a1++) {
        for( b1 = 0;  b1 < m;  b1++) {
BR          for( a2 = a1;  a2 < n;  a2++) {
                for( b2 = b1 ;  b2 < m;  b2++) (
                }
            }
    }
```

subarray

```
for( s = 0;  s < n;  s++) {
    for( e = s;  e n;  e++)
```

calculating the sum [a1, b1 to a2, b2] → prefix O(1)



Assume:

Total sub matrix = x

$$\sum_{i=0,\ j=0}^{i<n,\ j<n} count * mat[i][j]$$

1	3
2	8

⇒

a) 1 = 1

b) $1 \quad 3$ = 4

c) $\begin{matrix} 1 & 3 \\ 2 & 8 \end{matrix}$ = 14

d) 3 = 3

e) 2 = 2

f) 8 = 8

g) $\begin{matrix} 3 \\ 8 \end{matrix}$ = 11

h) $\begin{matrix} 1 \\ 2 \end{matrix}$ = 3

i) $2 \quad 8$ = 10

ans = 1 + 4 + 14 + 3 + 2 ---- 10

sum = $\underset{mat[0][0]}{1} * 4$ + 3 * 4 + 2 * 4 + 8 * 4 ⇒

4 ← count of submatrices where mat[0][0] is part of ?

count of occurences

	0	1	2	3	4	5
0	TL	TL	TL	TL		
1	TL	TL	TL	TL		
2	TL	TL	TL	TL (i,j) BR	BR	BR
3				BR	BR	BR
4				BR	BR	BR

TL : x = no of top left corners which will include mat[i][j]

BR : y = no of bottom right corners which will include mat[i][j]

Total combinations ⇒ $x * y$

$x \rightarrow (i+1) * (j+1)$

$y \rightarrow (n-i) * (m-j)$

```
int  calcSum( int[][]  mat) {
    int  n = mat.length;
    int  m = mat[0].length;
    int  sum = 0;
    for (i = 0; i < n;  i++) {
        for (j = 0; j < m;  j++) {
            int occ = (i+1) * (j+1) * (n-i) * (m-j)
            sum = sum + mat[i][j] * occ;
        }
    }
    return sum;
}
```

TC: $O(n * m)$

SC: $O(1)$

$O(n^6)$ ↓ $O(n^4)$ — prefix ↓ $O(n^2)$

Qu3. Given mat[n][m].

Row and col wise sorted matrix

find max submatrix sum

	0	1	2	3
0	-20	-16	-4	8
1	-10	-8	12	14
2	-1	6	21	30
3	5	7	28	42

Brute force: Go to all submatrices — $O(n^2 m^2) \simeq O(n^4)$

$O(1)$ — find sum of each submatrix &
update the max

TC: $O(n^4) \simeq O(n^2 m^2)$

SC: $O(n * m)$

```
TL  for( a1 = 0; a1 <n; a1++) {
        for( b1 = 0; b1 <m; b1++) {
BR          for( a2 = a1; a2<n; a2++) {
                for( b2 = b1 ; b2<m; b2++) {

                }
            }
        }
    }
```

	0	1	2	3
0	−20	−16	−4	8
1	−10	−8	12	14
2	−1	6	21	30
3	5	7	28	42

→ 42 must be part of submatrix with max sum.

1) 42 must be bottom right corner for every submatrix.

2.) fixed the bottom right corner

Code:

```
int maxsubmatrixsum(int[][] mat){
    int n = mat.length;
    int m = mat[0].length;

    int[][] pf = getPrefixsum(mat);

    int ans = -∞;

TL  for( a1 = 0; a1 < n; a1++){
        for( b1 = 0; b1 < m; b1++){
            // BR will always be (n-1, m-1)
    sum = calculate sum[(a1.b1) to (n-1.m-1)]
            ans = max(ans. sum);
        }
    }
    return ans;
}
```

Thankyou ☺