Arrays - 2

Q Given a binary string ('0', '1').
Maximize the total count of 1s by flipping any 1 substring!

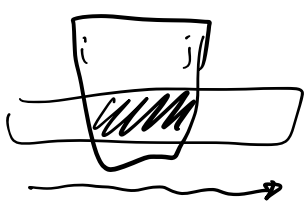
flip → 0 → 1
1 → 0

	0	1	2	3	4	5	6	7	8	9	10	11
S =	1	0	0	1	0	0	1	1	1	1	0	1

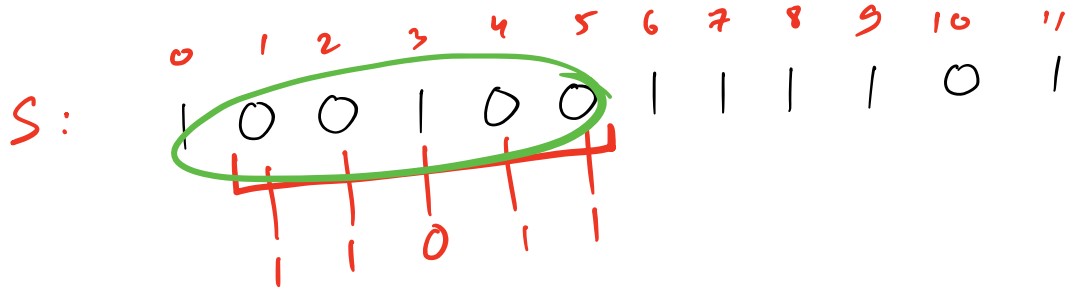
1 | 1 1 0 1 1 | 1 1 1 1 0 1 → 10

I) BF

```
ans = 0;
for( i = 0; i < N; i++) {
    for(j = i; j < N; j++) {
        flip(S, i, j);
        x = cntOne(s);
        ans = max(ans, x);
        flip(s, i, j);
    }
}
```



TC: O(N³)



Ones = 7

$7 - 1 + 4 = 10$ ✓

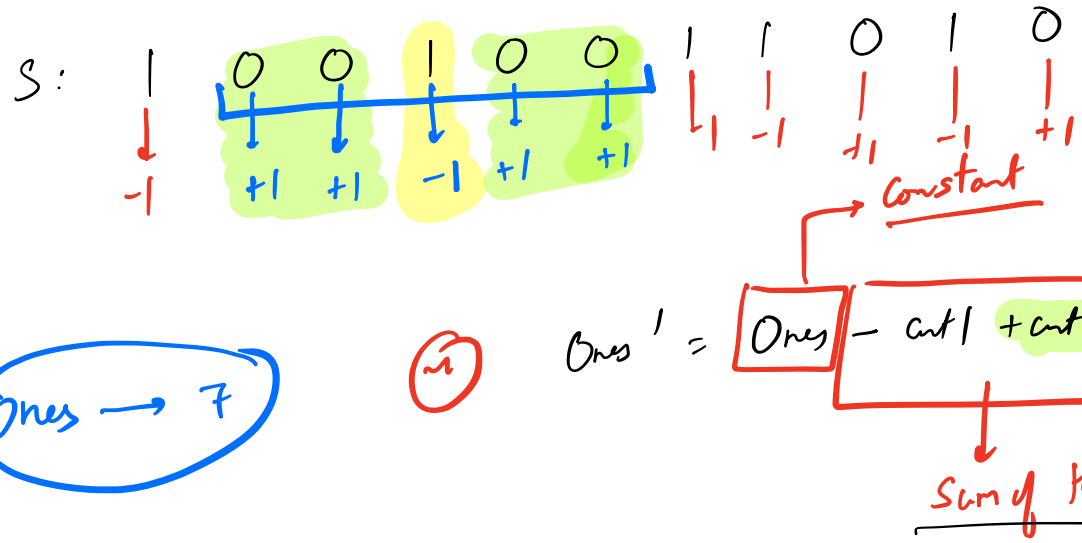
Substring: cnt0, cnt1

4 1

After flipping this S.S. # Ones = Ones − cnt1 + cnt0

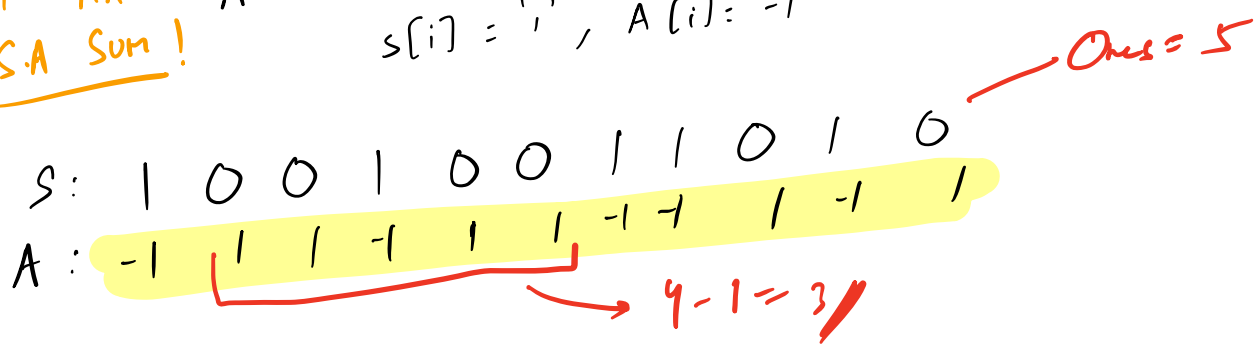
```
Ones = countOnes(S);
ans = 0
for(i=0; i<N; i++) {
    cnt0 = 0, cnt1 = 0;
    for(j=i; j<N; j++) {
        if(S[j] == '0') cnt0++;
        else cnt1++;
        ans = max(ans, Ones - cnt1 + cnt0);
    }
}
return ans;
```

TC: $O(N^2)$



Idea: find MAX S.A Sum!

S:
A: S[i] = '0', A[i] = +1
S[i] = '1', A[i] = -1



KADANE'S ALGO

Ones - cnt1 + cnt0

5 + 3 = 8

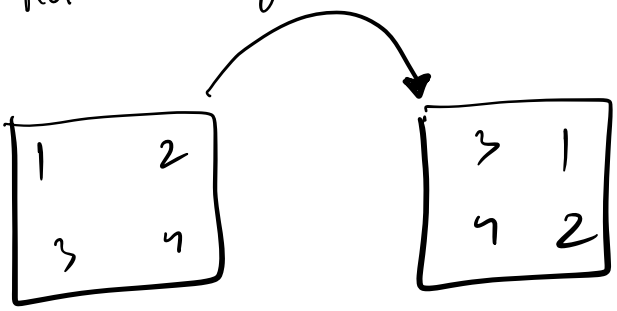
```
int maxSASum(A) {                          ──→ N
    sum = 0
    max = 0
    for (i=0; i<N; i++) {
        sum += A[i];
        if (sum < 0) {
            sum = 0
        }
        max = MAX(max, sum);
    }
    ret max;
}
```

```
ones = countOnes(S);                       ──→ N
S --'0'→1, '1'→-1--> A[]                   ──→ N
Sum = maxSASum(A);                         ──→ N
ret ones + sum;
```

TC : O(N)

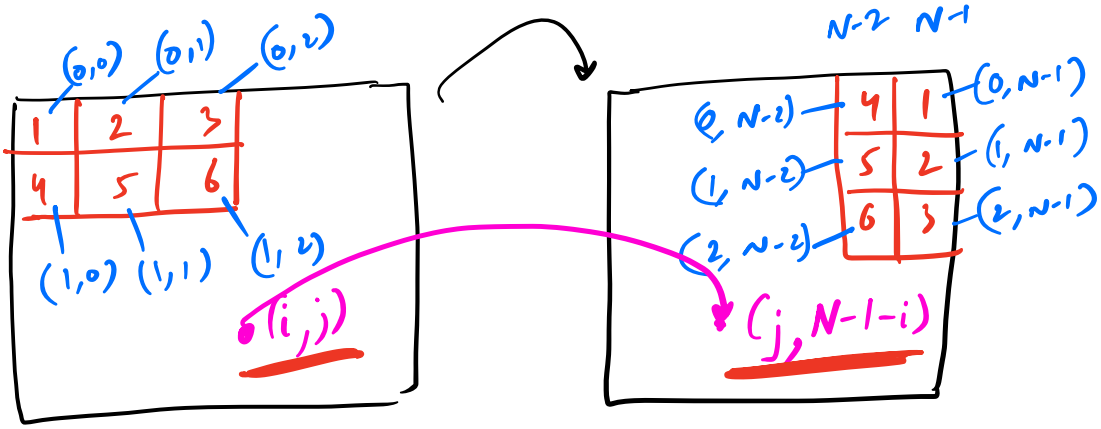
SC : O(N) → O(1) (don't need A[])

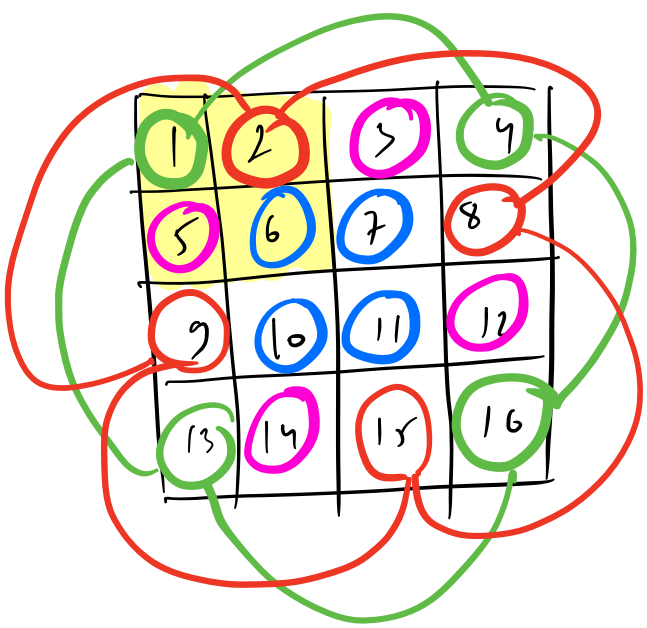
Q Given a N x N matrix.
Rotate it by 90° clockwise!

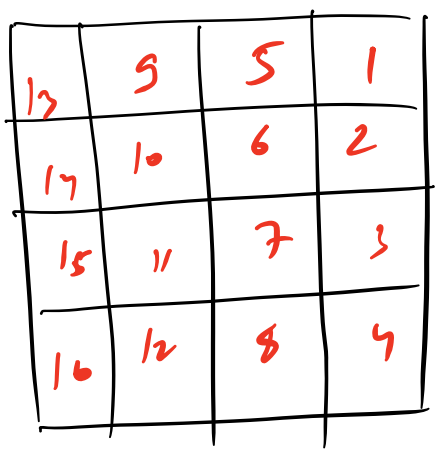


1) TRANSPOSE + REV ROWS! ✓

2)







TC: O(N²)

SC: O(1)

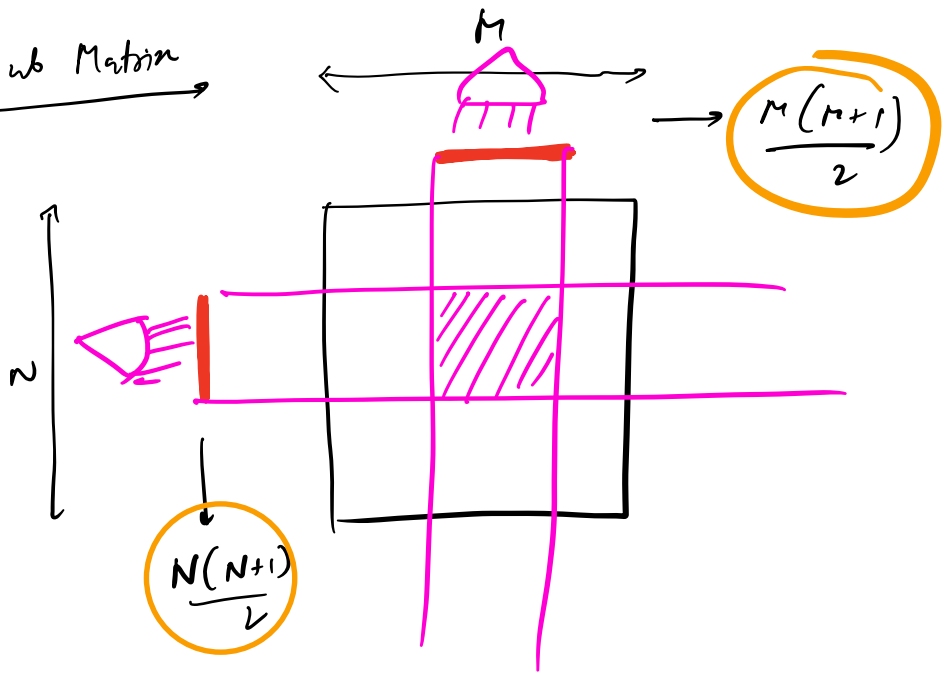
Q Given a 2D array.
Find the sum of all submatrices!

$$\begin{bmatrix} 3 & 6 \\ 5 & 4 \end{bmatrix}$$

[3] 9 [3 6] 15 [3 / 5] 14 [6 / 4] 10 [3 6 / 5 4] 24
[6] 6 [5 4] 9
[5] 5
[4] 4

= 96

(3) # Sub Matrix



$$\#SM = \frac{N(N+1)}{2} \times \frac{M(M+1)}{2} = \frac{NM(N+1)(M+1)}{4}$$

$O(N^2M^2)$

1) BF

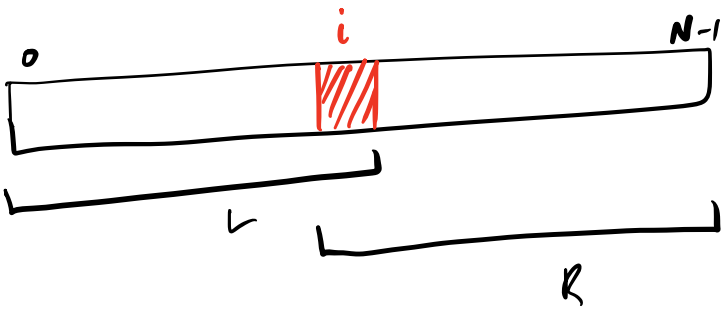
$\forall$ Sub Matrix $\longrightarrow$ N^2M^2

$\longrightarrow$ Cal Thur Sum $\longrightarrow$ $O(NM)$

$TC = O(N^3M^3)$

$\longrightarrow$ SAME PROBLEM [1D]

Contribution Technique!



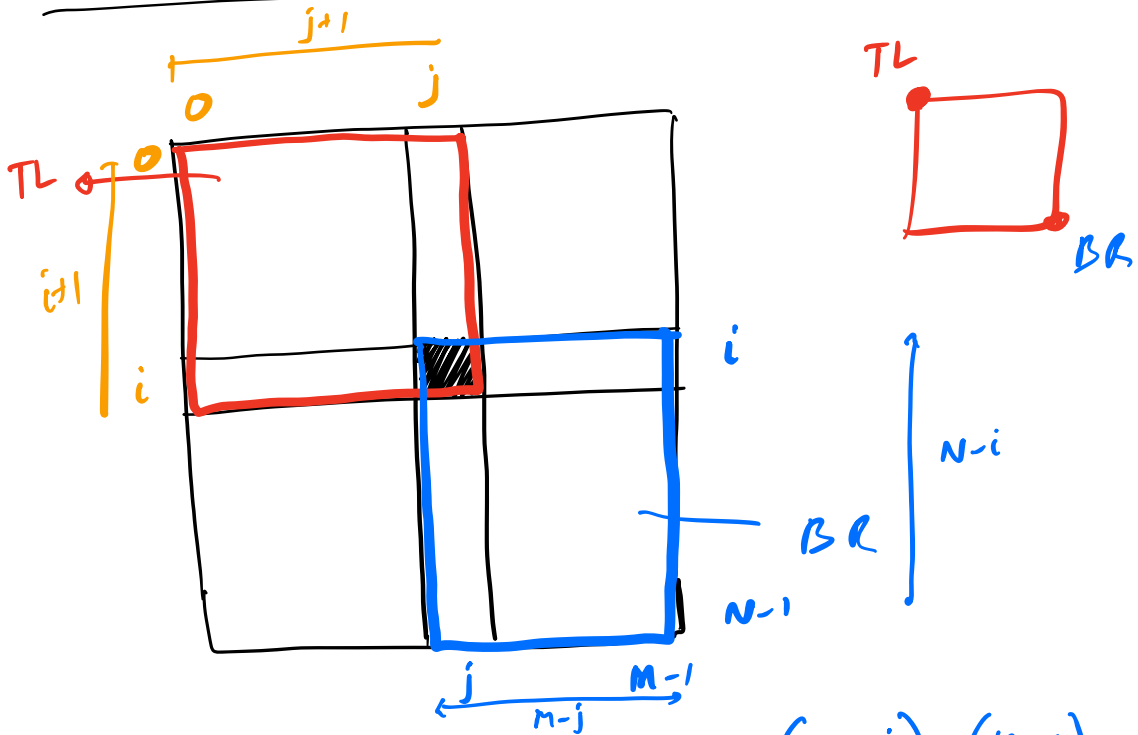
$(i+1) \times (N-i)$

$$ANS:- \sum_{i=0}^{N-1} A[i] \times [(i+1) \times (N-i)]$$

$TC = O(N)$

$SC = O(1)$

○ Come back to 2D problem



TL = $(i+1) \times (j+1)$ # BR = $(N-i) \times (M-j)$

SMs containing i^{th} element

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

$$ANS = \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} A[i][j] \times (i+1)(j+1)(N-i)(M-j)$$

TC = O(NM)

SC = O(1)

Q Given a 2D array & q queries.
Find the sum of the submatrix defined in the query!

[TL_x, TL_y, BR_x, BR_y]

A:

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

q

[0, 0, 1, 3] → 18

[1, 1, 1, 1] → 1

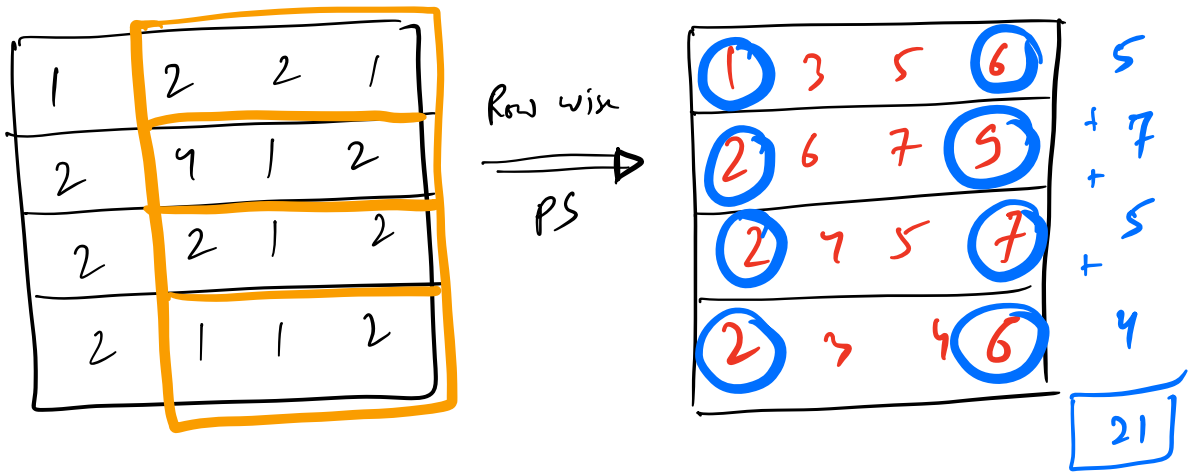
[0, 0, 2, 3] → 28

BF

∀ q → iterate on the submatrix & find sum. → O(NM)

TC: O(NMq)

1) Prefix Sum (Rows)



Build PS → O(NM)

+

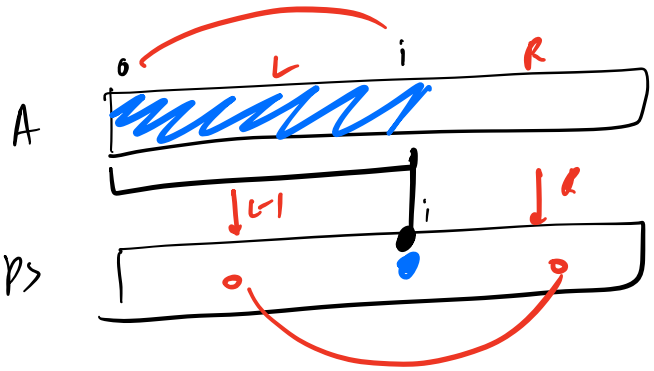
1 query → O(N)

Q → O(Q.N)

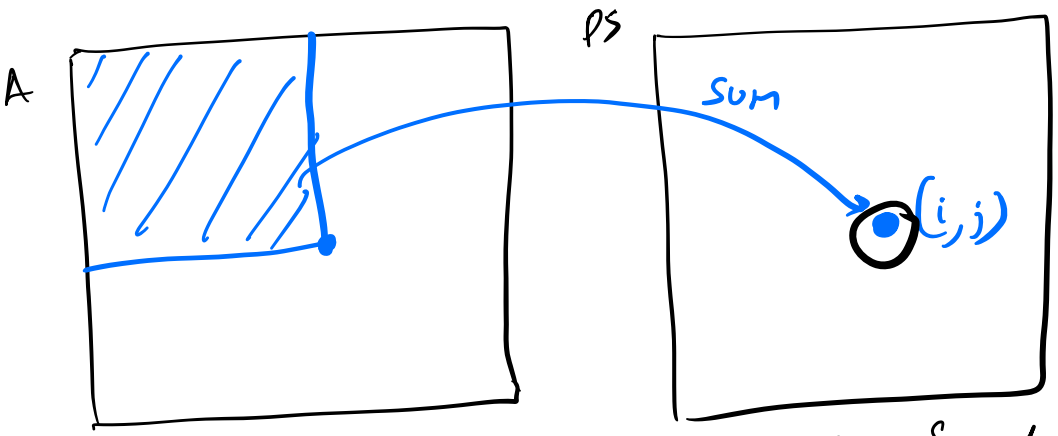
TC = O(N(M+Q))

SC = O(NM) → O(1) [SAME MATRIX]

①



$PS(i) = 0 - -i$



PS[i][j] → Sum of the SM((0,0),(i,j))

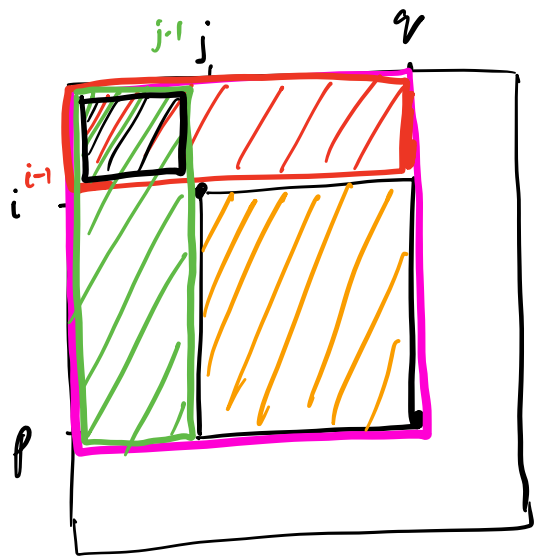
Ex:

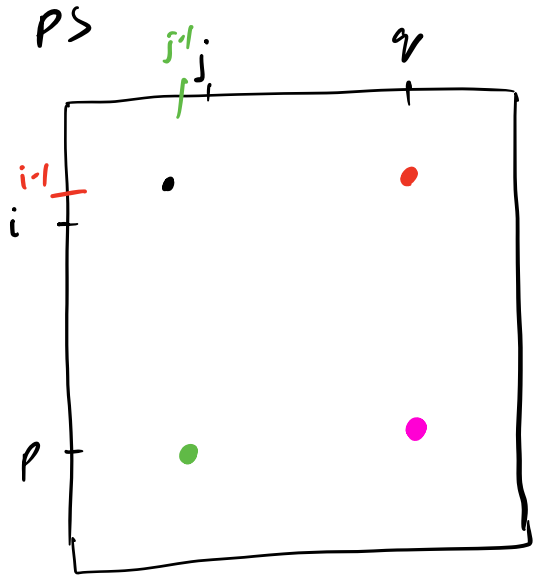
A

1	2	1
4	2	1
1	2	3

PS

1	3	4
5	9	11
6	12	17





ORANGE = PINK − RED − GREEN + BLACK

$$sum((i,j),(p,q)) = PS[p][q] - PS[i-1][q] - PS[p][j-1] + PS[i-1][j-1]$$

1 $q_1 \rightarrow O(1)$

q $q_1 \rightarrow O(q)$

BUILD 2D PS →

A

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

O(NM) → Row wise PS

$$\begin{bmatrix} 1 & 3 & 6 & 10 \\ 1 & 3 & 4 & 5 \\ 2 & 3 & 5 & 6 \\ 3 & 4 & 5 & 6 \end{bmatrix}$$

↓ col wise PS ↓ O(NM)

2D PS →

$$\begin{bmatrix} 1 & 3 & 6 & 10 \\ & 6 & 10 & 15 \\ 2 & & 9 & 15 & 21 \\ 4 & & & \\ 7 & 13 & 20 & 27 \end{bmatrix}$$

Build PS → O(NM)

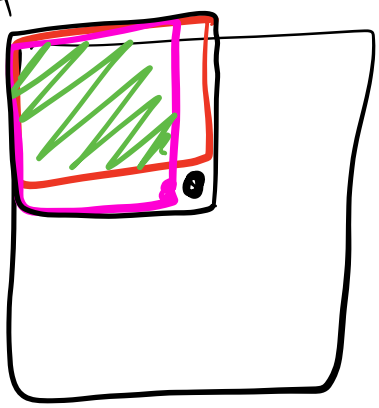
1 Qn → O(1)

Q Qn → O(Q)

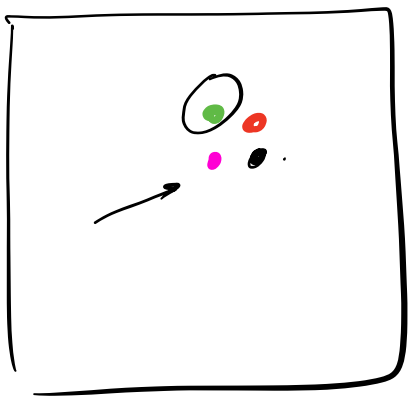
TC = O(NM + Q)

SC = O(NM) → O(1) [SAME ARRAY]

A



PS



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