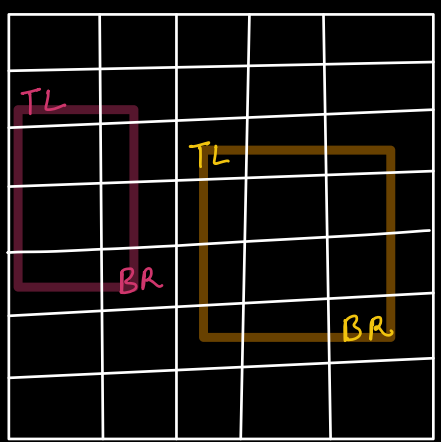
Q Given a matrix of size N x M & Q queries.
Every query gives boundries of a sub matrix.
Return the sum of the sub-matrix.

Sub Matrix Sum Queries




Q queries.

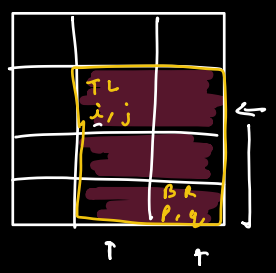
| TL | BR |
|---|---|
| (i, j) | (p, q) |

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

| TL | BR | |
|---|---|---|
| (1, 2) | (4, 3) | ⇒ 20 |
| (1, 1) | (1, 1) | ⇒ 5 |
| (5, 1) | (5, 3) | ⇒ 7 |

## Brute Force

TL (i, j)  BR (p, q)

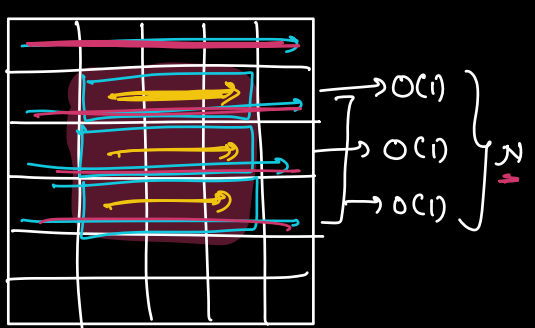



for every query { ⟹ Q

N×M { sum = 0;
for (row = i; row <= p; row++) {
for (col = j; col <= q; col++) {
sum += A[row][col];
}
}

TC : O(Q N M)

## Row/col Prefix sum approach




→ Convert every row to prefix sum

⇒ O(NM) + O(QN) .

if N >> M

→ Convert every col to prefix

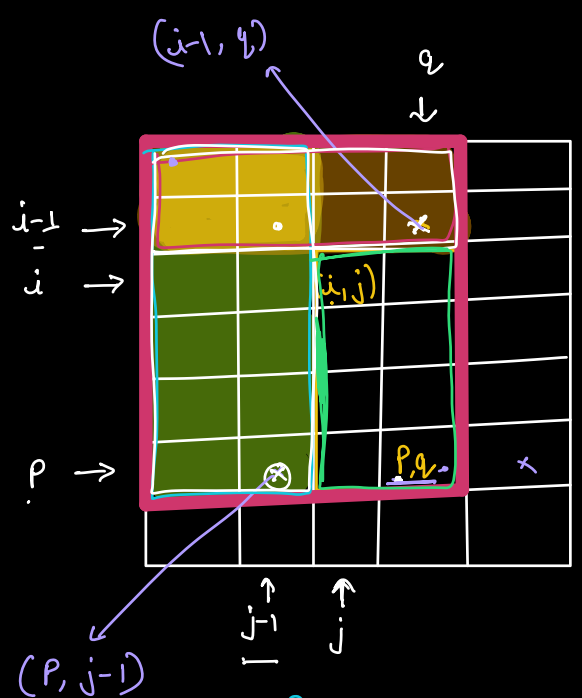
O(NM) + O(QM)

SC : O(NM)

for N×N ⟹ O(NM) + O(QN)

PS[i] → sum of all elements from index 0 to i
(in 1D array) (s) (e)

⇒ PS[i][j] → sum of all elements in the submatrix
(for 2D array) from 0, 0 (TL)
to i, j (BR)

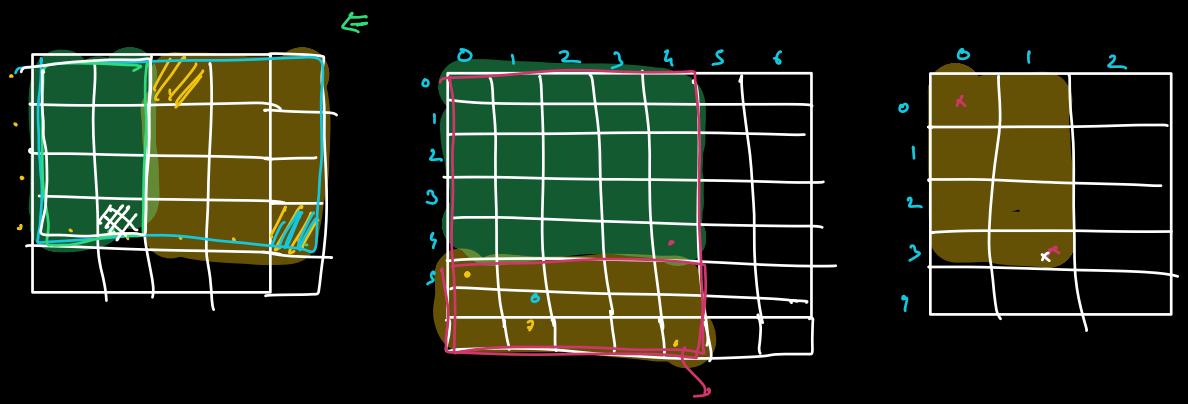



TL BR
⇒ (i,j) (p, q)

PS[p][q] ⇒ Sum of submatrix from (0,0) to (p,q)

Sum from (i,j) to (p,q) = PS[p][q] - PS[p][j-1] - PS[i-1][q] + PS[i-1][j-1]

| TL | BR | | |
|---|---|---|---|
| (1,2) | (3,5) | ⇒ | PS[3][5] - P[0][5] - PS[3][1] + PS[0][1] |
| (0,2) | (3,4) | ⇒ | Ps[3][4] - P[3][1] |
| (5,0) | (7,4) | ⇒ | PS[7][4] - Ps[4][4] |
| (0,0) | (3,1) | ⇒ | PS[3][1] |



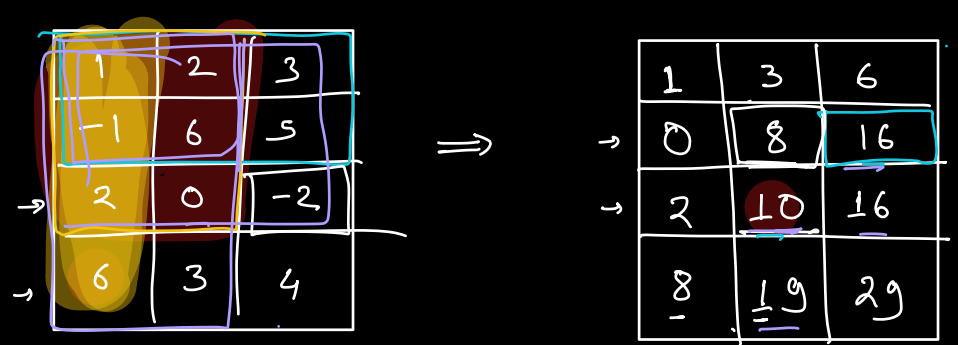
if $i == 0$

Sum $(i,j)(p,q)$ $\longrightarrow$ $PS[p][q] - PS[p][j-1]$

if $j == 0$

Sum $(i,j)(p,q)$ $\longrightarrow$ $PS[p][q] - PS[i-1][q]$

## How to build this PS matrix ?




16 + 10 - 8 - 2

16 + 19 - 10 + 4

| | | |
|---|---|---|
| $a_0$ | $b_0$ | $c_0$ |
| $a_1$ | $b_1$ | $c_1$ |
| $a_2$ | $b_2$ | $c_2$ |

PS Mat ⟹

| | | |
|---|---|---|
| $a_0$ | $a_0+b_0$ | $a_0+b_0+c_0$ |
| $a_0+a_1$ | $a_0+b_0$<br>$a_1+b_1$ | $a_0+b_0+c_0$<br>$a_1+b_1+c_1$ |
| $a_0+a_1+a_2$ | $a_0+b_0$<br>$a_1+b_1$<br>$a_2+b_2$ | $a_0+b_0+c_0$<br>$a_1+b_1+c_1$<br>$a_2+b_2+c_2$ |

Row PS ↓

| | | |
|---|---|---|
| $a_0$ | $a_0+b_0$ | $a_0+b_0+c_0$ |
| $a_1$ | $a_1+b_1$ | $a_1+b_1+f$ |
| $a_2$ | $a_2+b_2$ | $a_2+b_2+c_2$ |

Col PS ↑

## TC of solving Q queries

$O(MN) + O(Q)$

$O(MN)$ ↑ Building PS matrix

$O(Q)$ ↑ Solving Q queries.

Q Given a matrix. Find sum of all possible submatrix. (Elementary)

(N×N)

Google
Facebook

$$\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$$

4×1

$$[1] + [1] + [1] + [1]$$

$$\begin{bmatrix} 1 \\ 1 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$[1 \quad 1]$$
$$[1 \quad 1]$$

$$\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$$

---

16

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \Rightarrow 500$$

Break till 10:36 pm

**Brute Force** — Iterate over all submatrix.

↳ Pairs of TL & BR

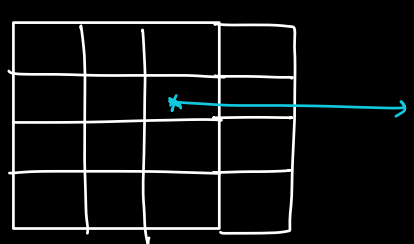
```
sum = 0;
for(i=0; i<N; i++){
    for(j=0; j<N; j++){                    } O(N^2)
        // TL -> i, j
        // for TL iterate over all BR
        for(p=i; p<N; p++){
            for(q=j; q<N; q++){            } O(N^2)
                // BR -> p, q
                // find sum for (i,j) to (p,q)
                for(r=i; r<=p; r++){
                    for(c=j; c<=q; c++){   } O(N^2)
                        sum += A[r][c];
                }
            }
        }
    }
}
```

TC: $O(N^6)$

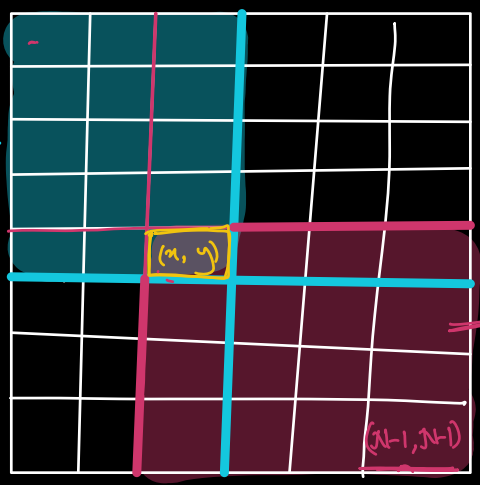
↓

$O(N^4)$

Using PS matrix



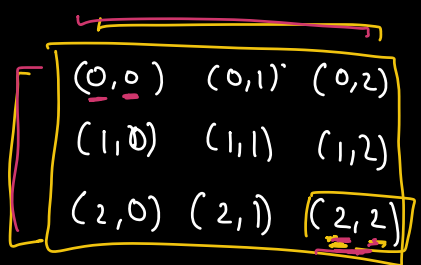
Count in how many submatrix this element will be present




No of cells
$= (x+1) \times (y+1)$
(TL)

No of cells
$(N-x) \times (N-y)$
(BR)

N x N



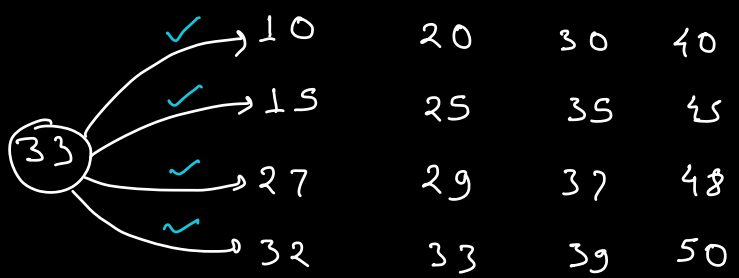

```
Sum = 0;
// iterate over all elements of matrix
// count the contribution
for (i=0; i<N; i++) {
    for (j=0; j<N; j++) {
        TL = (i+1) x (j+1);
        BR = (N-i) x (N-j);
        SubMatrix cnt = TL x BR
        Sum = Sum + SumMatrix cnt x A[i][j];
    }
}
```

TC : $O(N^2)$

SC (Extra) : $O(1)$

Q Amazon MS

Given a N×N matrix where all rows are sorted & all cols are sorted. Given a no K. Return true if K exists in the matrix.

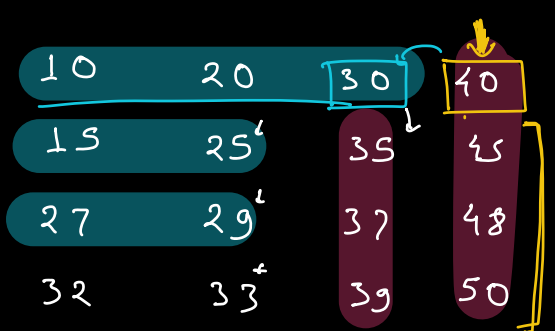



K = 35 → True
K = 28 → False
K = 33

TC: $O(N^2)$ (Brute force)

Binary Search ⇒ Apply BS on every row.

$O(N \log N)$



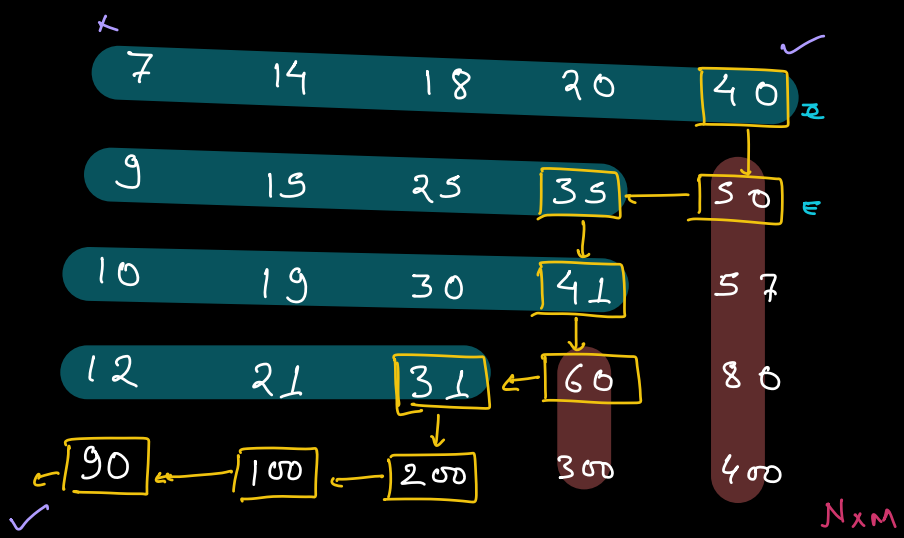

Min element → (0,0)
Max element → (N-1, N-1)

K = 33

Curr = TR corner

if (target < curr ele)
⇒ Discard curr col.

else if (target > curr ele)
Discard the row.




target - 42 ✗

```
i = 0 , j = N-1;
while ( i<N && j>=0 ) {
  if ( A[i][j] == target)
      ret true;
  if ( A[i][j] > target ){
    // discard col j
  }   j--;

  else if ( A[i][j] < target) {
    // discard row i
      i++;
  }

}
ret false;
```

TC : O(N+M)

SC : O(1) (Extra)

81 ⇒ 18

| | | | | |
|---|---|---|---|---|
| 7 | 14 | 18 | 20 | 40 |
| 9 | 15 | 25 | 35 | 50 |
| 10 | 19 | 30 | 41 | 57 |
| 12 | 21 | 31 | 60 | 80 |
| 90 | 100 | 200 | 300 | 400 |

33

| | | | | |
|---|---|---|---|---|
| 1 | 10 | 20 | 30 | 40 |
| 2 | 15 | 25 | 35 | 45 |
| 3 | 27 | 29 | 37 | 48 |
| 4 | 32 | 33 | 39 | 50 |

27

$$|x| = \begin{cases} -x & \text{if } x < 0 \\ x & \text{if } x \geq 0 \end{cases}$$

$$|-2| = -(-2) = 2$$

$$|2| = 2$$