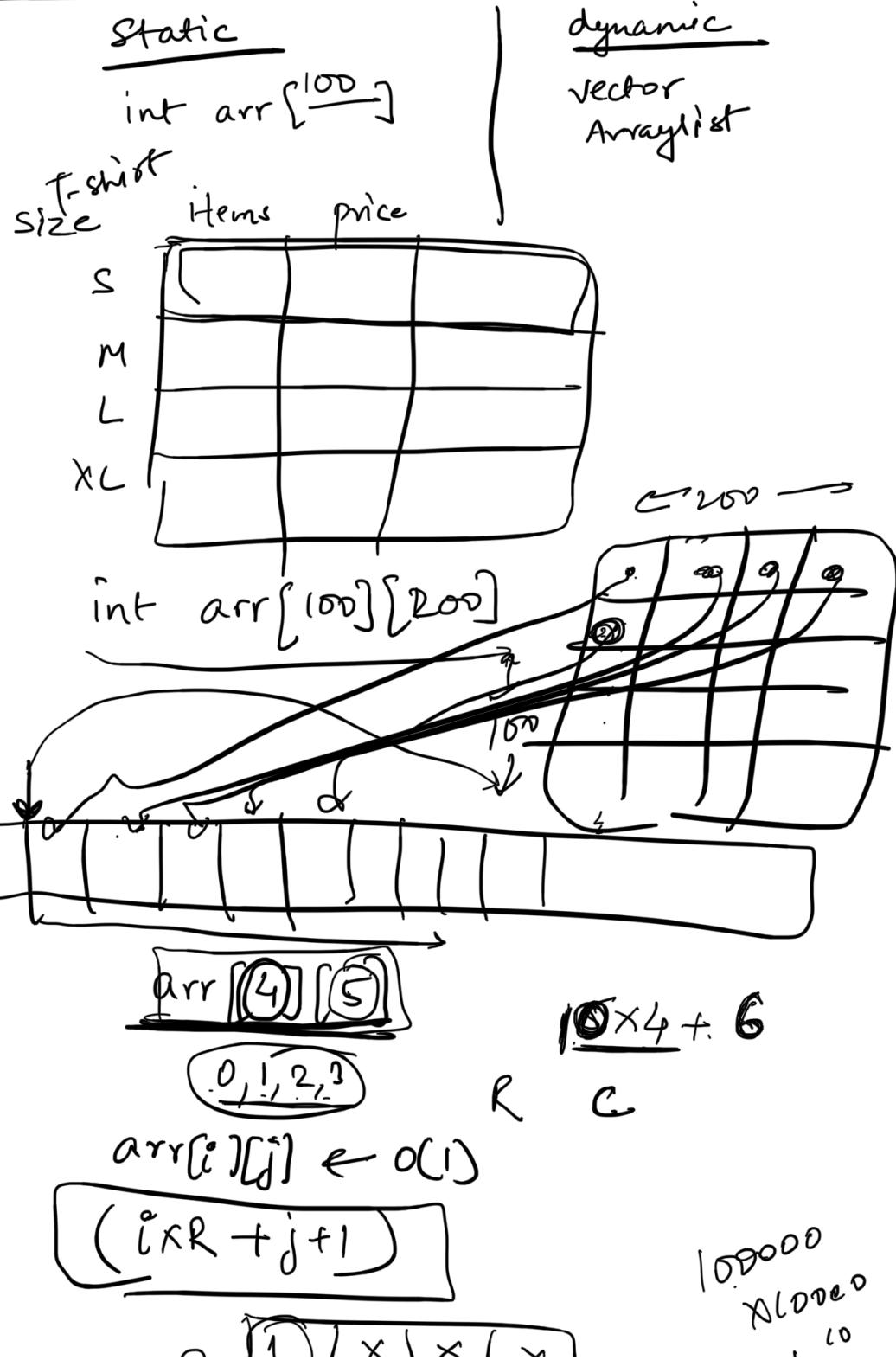


ARRAY AND MATH - II

- 2-D arrays (Multi-D)
- Maths

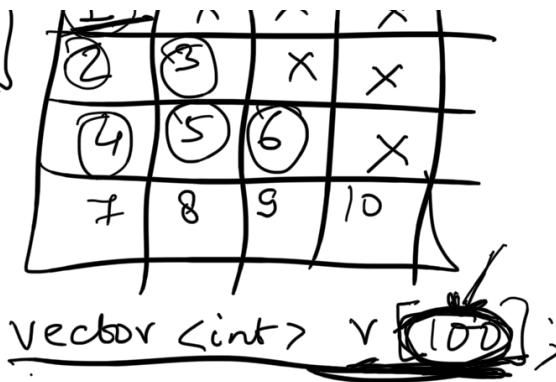


int arr[R][C]

$v[0] \rightarrow 2^{13}$
 $v[1] \rightarrow 1$
=

②

→ L



= 10

$v[i] \rightarrow i^{\text{th}}$
 $v[i]$

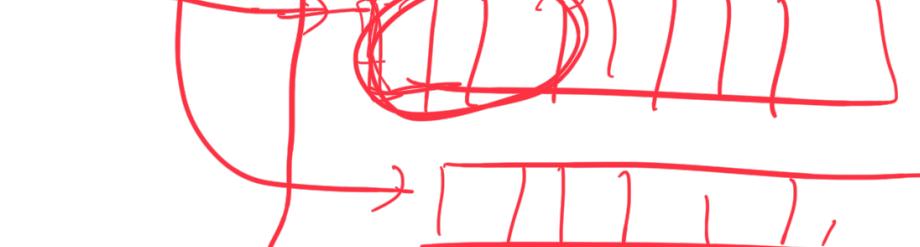
vector < int > v[100];

vector < vector < int > > twoD-v.

vector < int > newRow;
newRow.push-back(1);
twoD-v.push(newRow);

newRow2

↳ (2, 3)



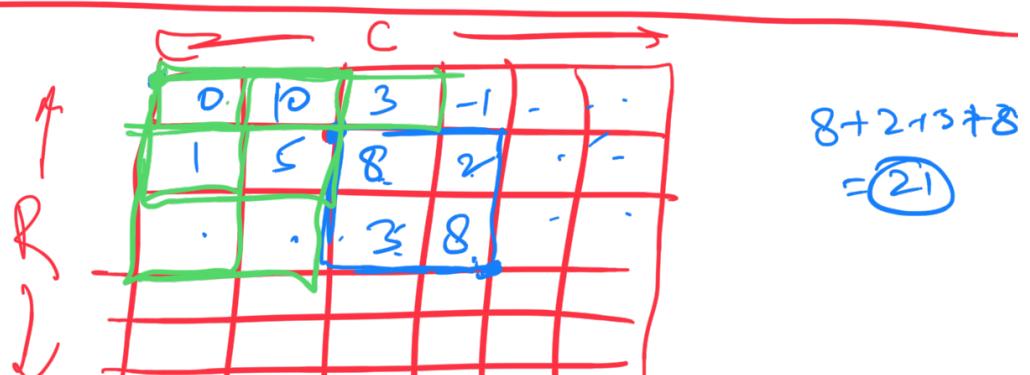
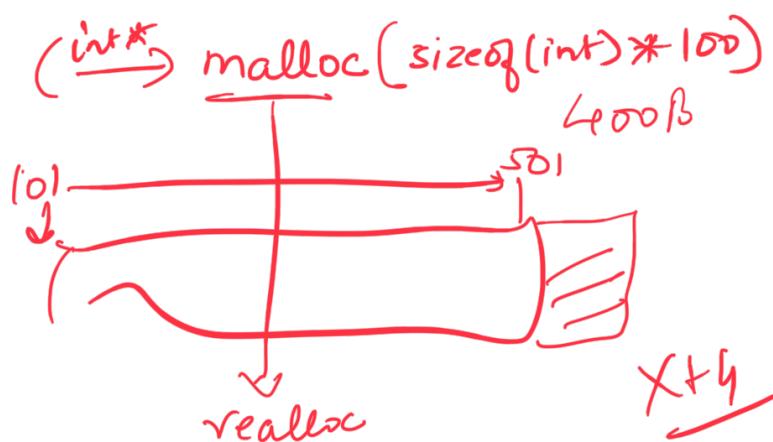
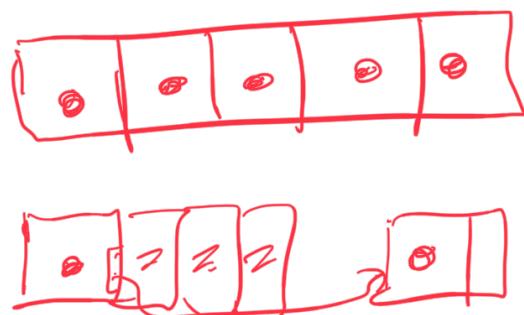
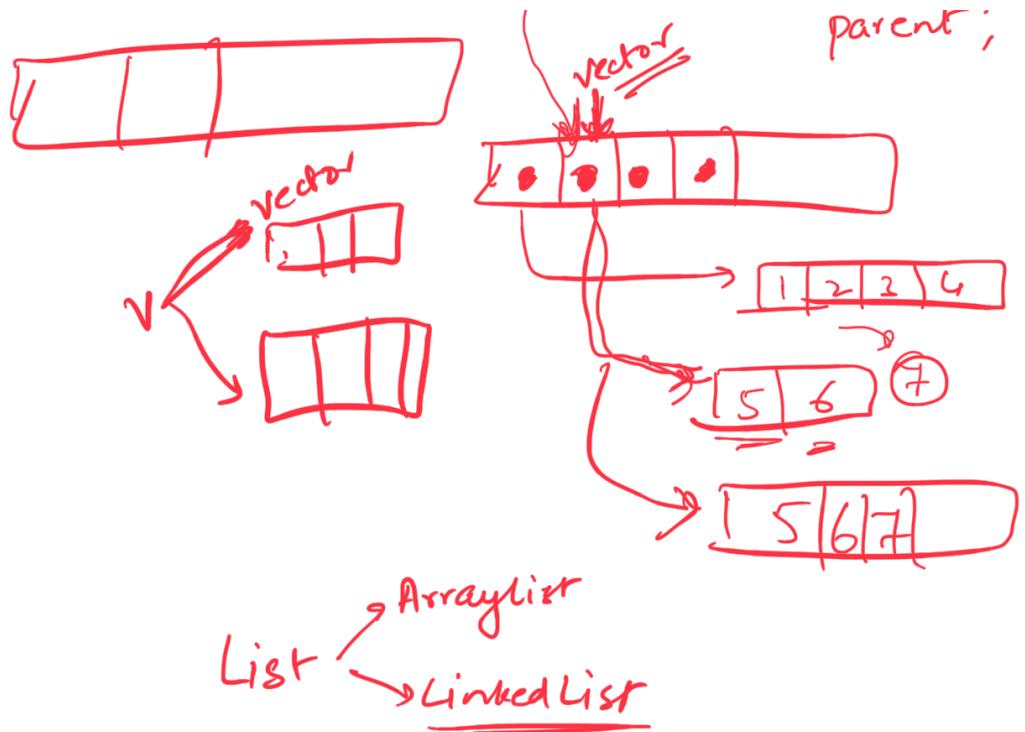
Vector < int >

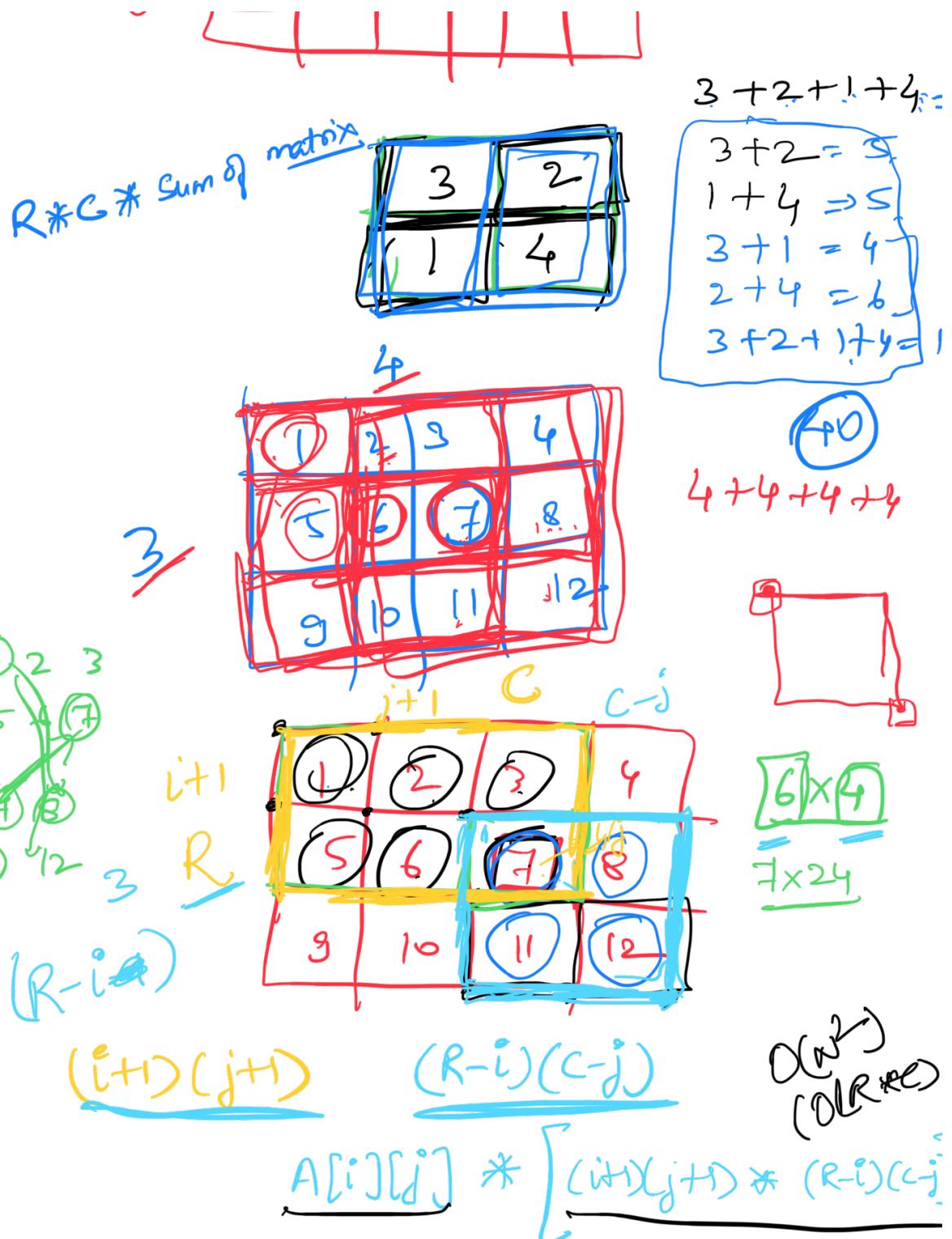
class

}

$v[0][0]$

Vector < vector < int > >





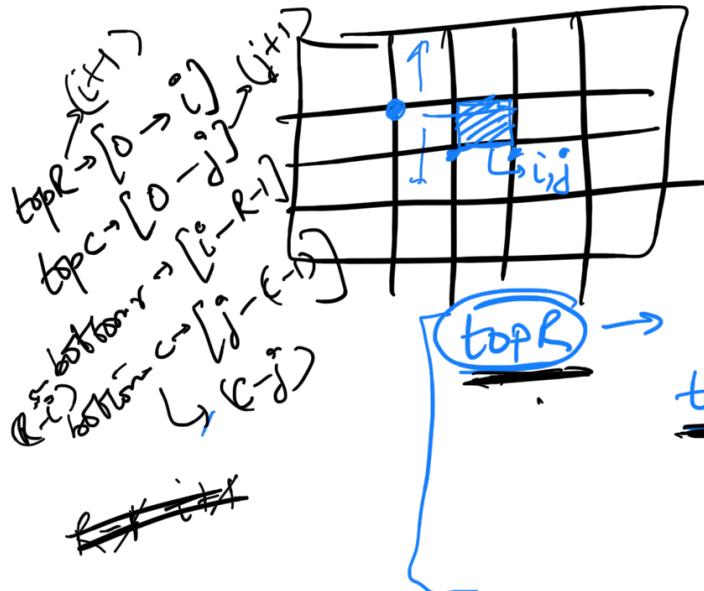
```
for (int topv1 = 0; topv2 < R; topv2++)
    for (int topcl = 0; topcl < C; topcl++)
        for (int bottomv1 = topv1;
             bottomv1 < R; bottomv1++)
            for (int bottomcl = topcl;
                 bottomcl < C; bottomcl++)
```

$O(R^3 C^3)$
 $O(N^6)$

$\frac{topv1, topcl, bottomv1, bottomcl}{topv1^2}$

$\text{for } (i = \frac{topv1}{topv1}, i < bottomv1; i++)$

for $i = \text{topC1}; j < \text{bottomC1}$
 $\text{sum} += A[i][j]$



$\text{topR} \leq i \leq \text{bottomR}$
 $\text{topC} \leq j \leq \text{bottomC}$
bottom right

O to R :

topC → O to C :

bottomR → topR to R :

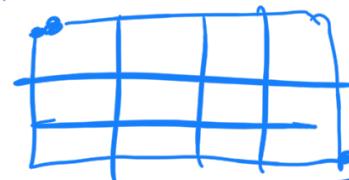
bottomC → topC to C :

R → topR to bottomR
C → topC to bottomC

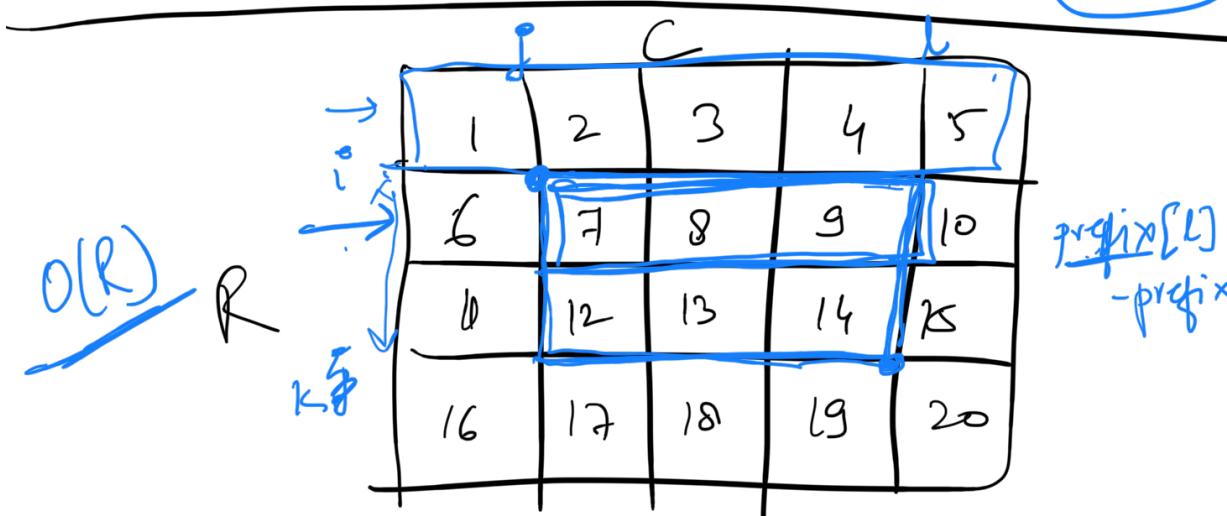
$\text{sum} += A$

O(R)

topR, topC



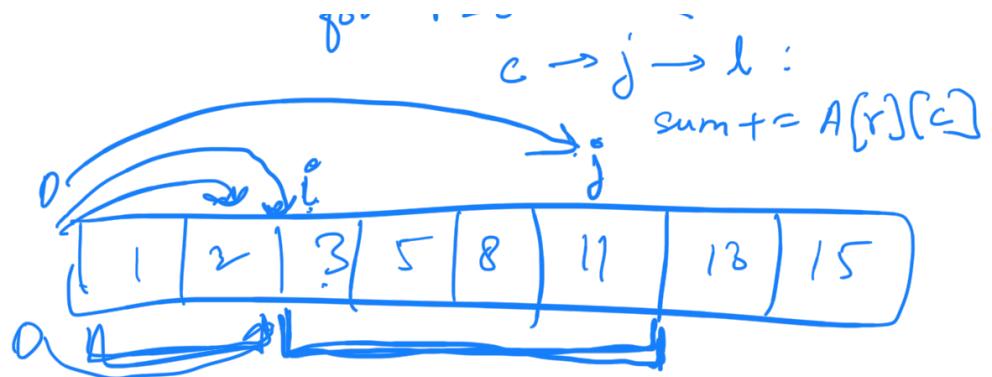
bottomR, bottomC



prefix[L]
-prefix

Query → (i, j) → (k, l)

for $r = i \rightarrow k$



prefix sum

$$\frac{1+2+3+5+8+11}{\cancel{1} \cancel{2} \cancel{3} \cancel{5} \cancel{8} \cancel{11}}$$

$$\begin{array}{c} \boxed{\text{prefix}[i]} \rightarrow \boxed{0 \text{ to } i} \\ \hline \boxed{i \text{ to } j} \rightarrow \boxed{\text{prefix}[j] - \text{prefix}[i]} \end{array}$$

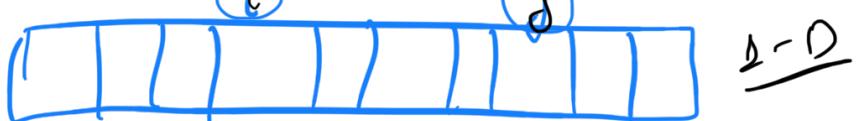
$$\text{prefix}[0] = A[0];$$

$$\text{prefix}[1] = \underline{A[0] + A[1]}$$

$$\text{prefix}[2] = \underline{A[0] + A[1] + A[2]}$$



$$\text{prefix}[i] = \underline{\text{prefix}[j-1] + A[i]},$$

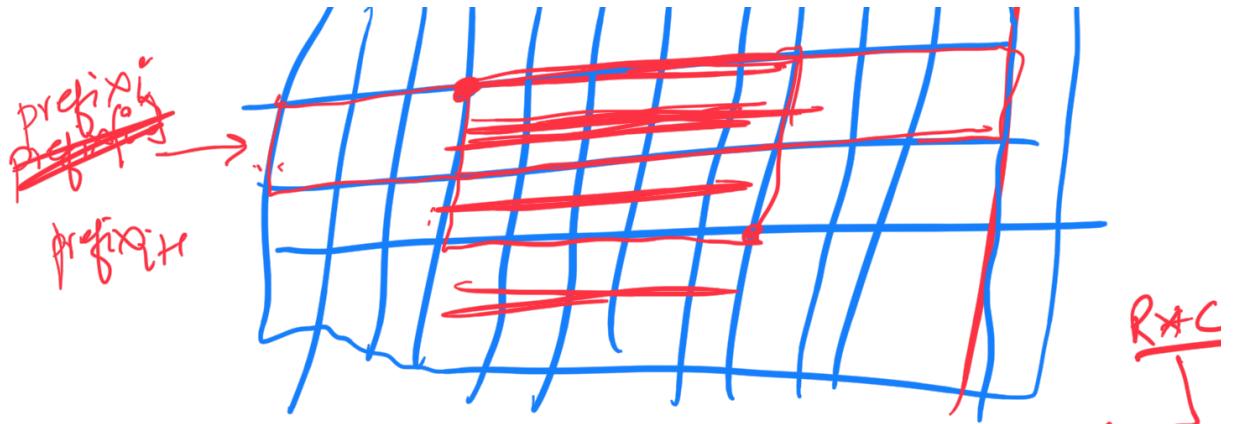


$$\text{prefix}[j] = \underline{A[0] + A[1] + A[2] + \dots + A[i]} + A[j]$$

$$\text{prefix}[i] = A[0] + \dots + A[i-1]$$

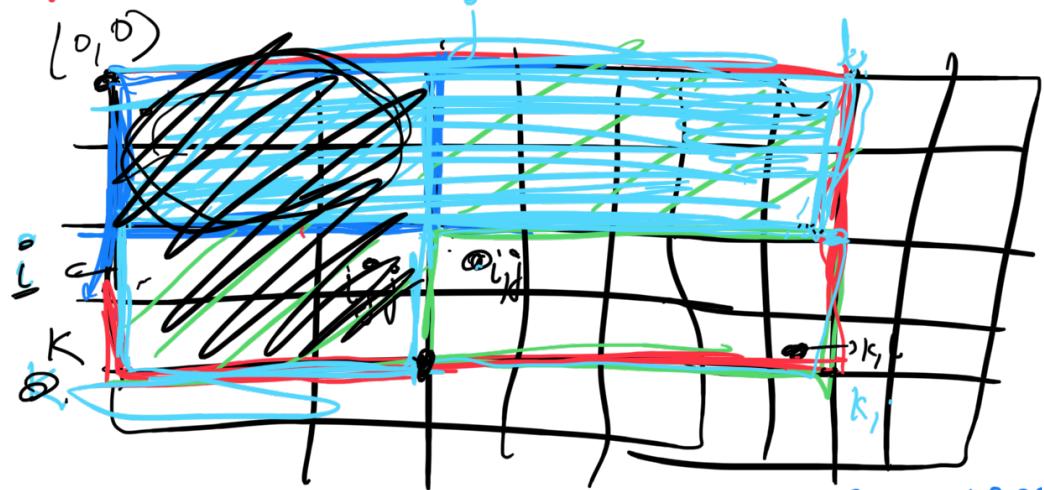
$$\text{prefix}[j] - \text{prefix}[i-1]$$

Diagram illustrating the computation of prefix sums for an array A with elements $1, 2, 3, 5, 8, 11, 13, 15$. The array is indexed from 0 to 7. A pointer i points to index 3 (value 5), and a pointer j points to index 5 (value 11). A curved arrow labeled $i \rightarrow j$ indicates the range from index 3 to 5. The formula $\text{prefix}[j] - \text{prefix}[i-1]$ is shown, where $i-1$ is the index of the element to be subtracted (4) and j is the index of the current element (5).



$\text{prefix}_i[l] - \text{prefix}_i[k-1]$

$O(R)$



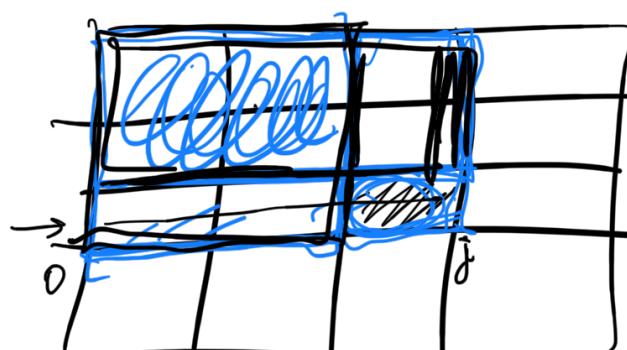
$$\begin{aligned} \text{prefixSum}[i][j] &= A[0, 0] + A[0, 1] + \dots + A[0, j] \\ &+ A[1, 0] + A[1, 1] + \dots + A[1, j] \\ &+ \dots \\ &+ A[i, 0] + A[i, 1] + \dots + A[i, j] \end{aligned}$$

$\Rightarrow O(C)$

$\text{prefixSum}[i-1][j]$

$\text{prefixSum}[k][l] - \text{prefixSum}[i-1][l]$

$- \text{prefixSum}[k][j-1] + \text{prefixSum}[i-1][j-1]$



$$\text{prefixSum}[i][j] = \text{prefixSum}[i][j-1] + \text{prefixSum}[i-1][j] - \text{prefixSum}[i-1][j-1] + A[i][j]$$

1	2	3	4
5	6	7	8
9	10	11	12

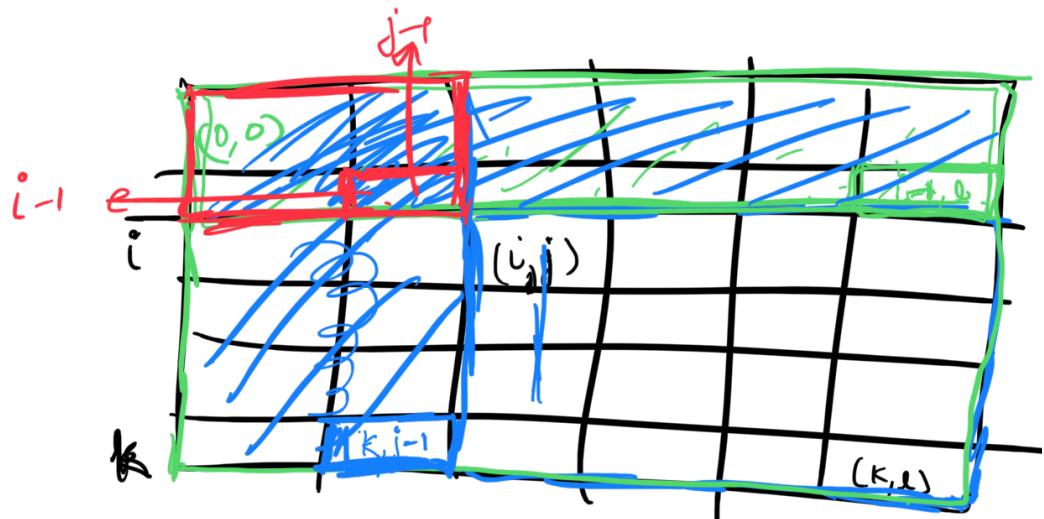
$$\text{prefixSum}(\cdot)[j]$$

8

pre processing time + $Q * O(\text{queries})$

$$R * C * O(1) + Q * O(1)$$

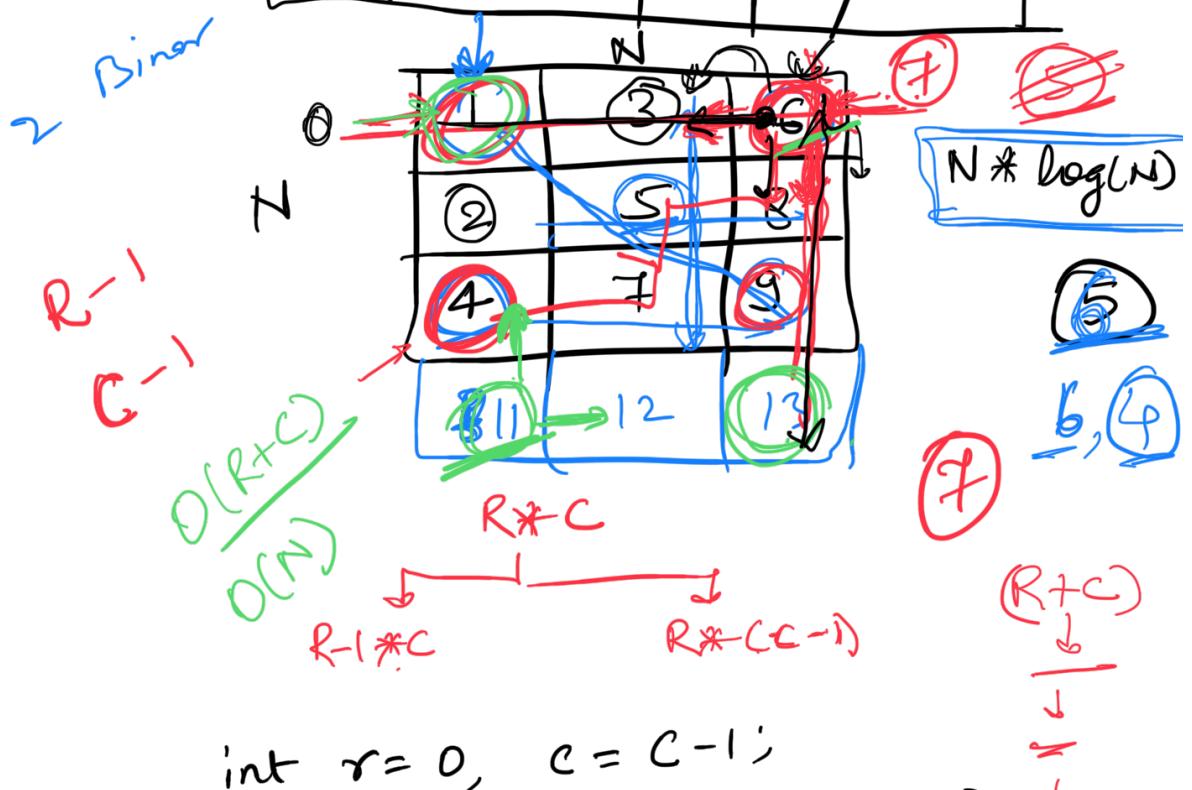
$$O(R * C + Q)$$



$$\text{prefixSum}(k, l) - \text{prefixSum}(i-1, l) - \text{prefixSum}(k, j-1) + \text{prefixSum}(i-1, j-1)$$

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

5	50	500	5000	50000	500000
49	51	5001	50000	500000	5000000



int $r = 0, c = C-1;$

while ($r < R \text{ & } c \geq 0$) {

 if ($\text{key} == A[r][c]$)
 return 1;

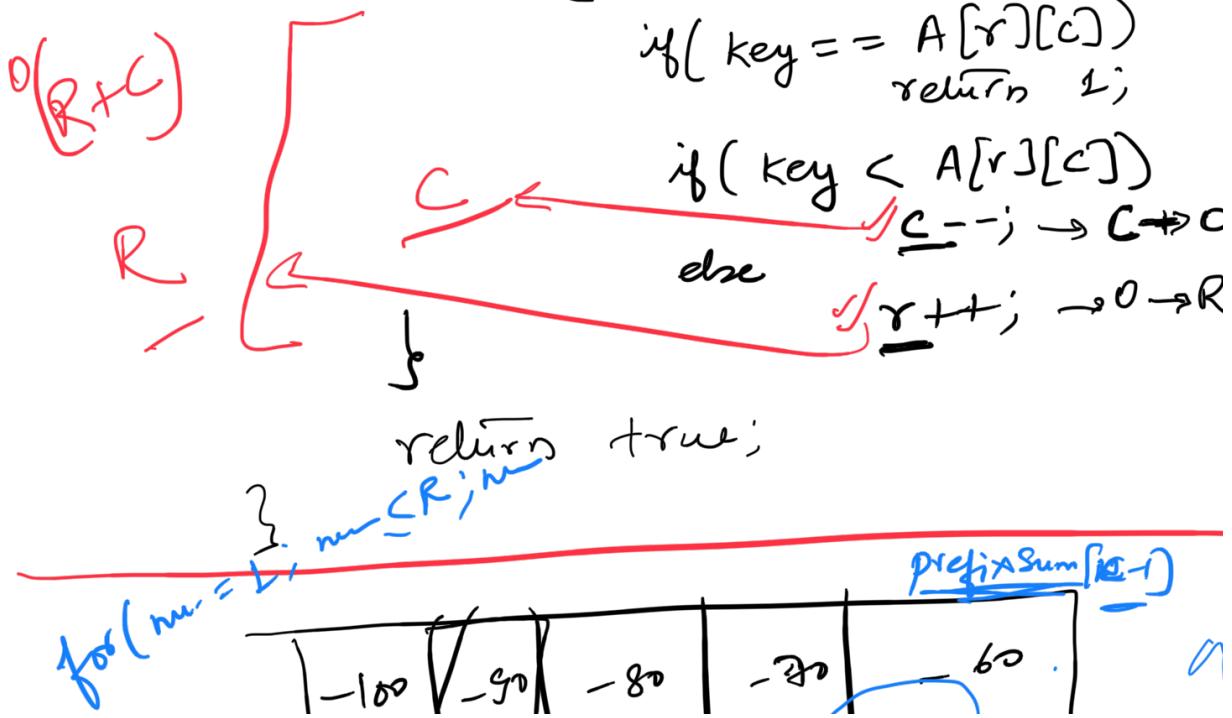
 if ($\text{key} < A[r][c]$)
 $c-- \rightarrow C \rightarrow c$

 else

$r++ \rightarrow 0 \rightarrow R$

return true;

$\{ m < R \}$



		-90	-80	-70	-60	-50	
	3	-50	0	4	5	6	
2	2	10	11	12	13	100	
1	1	10	11	12	13	100	
		10	11	12	13	100	

Max Sum Submatrix

$(i, j) \rightarrow \cancel{(R-1, C-1)} \rightarrow \text{Suffix sum}$

$(0, 0) \text{ to } (i, j)$ $O(R \times C)$

Time: $O(R \times C)$

Space: $O(R \times C)$

114	9	1	4	2
-14	-8	-7	-6	100
2	4	5	6	100
3	5	6	7	100
114	9	11	14	

max submatrix $\rightarrow L, 2, ?$

114
139
13

-14	-12
-11	-5

-5

L P