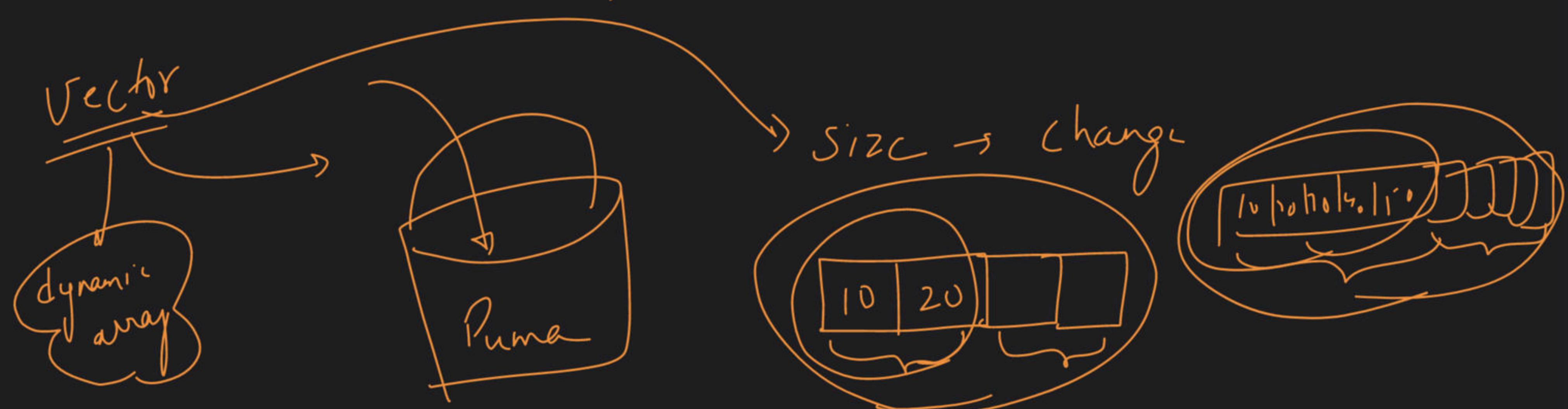
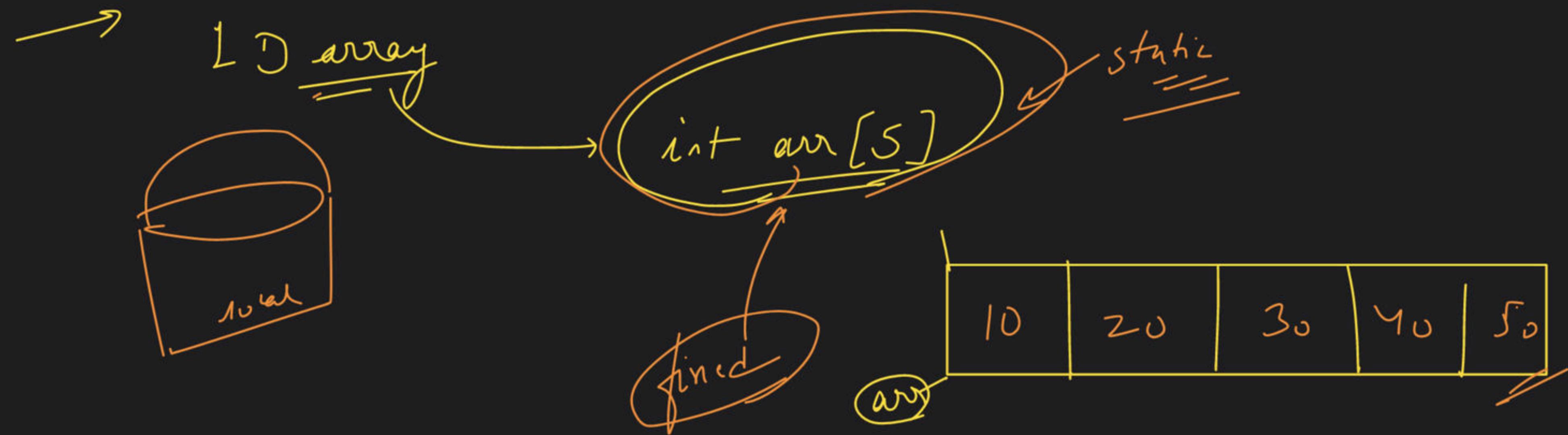


Array Class-3

Special class

S1 L

Vector → video
↓
Quality - medium



vector

→ declare

remove

arr.pop-back()
arr.pop-back()

vector < int > arr;

vector < int > arr(10)

vector < int > arr(10, -1)

pos, size

insert

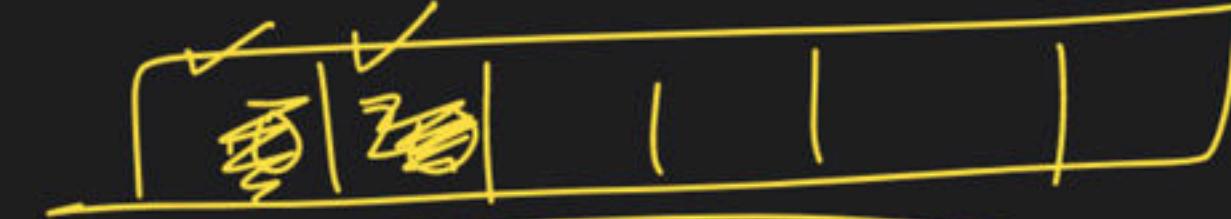
(I) → size define → access → same like array

arr[3] = 40

(II) → size not
define

→ insert →

arr.push-back(10)
arr.push-back(20)



Traversal

vector <int> arr;

arr.push_back(10)

(20)

(30)

(70)

int target = 70;

arr

10	20	30	40
0	1	2	3

int n =

arr.size()

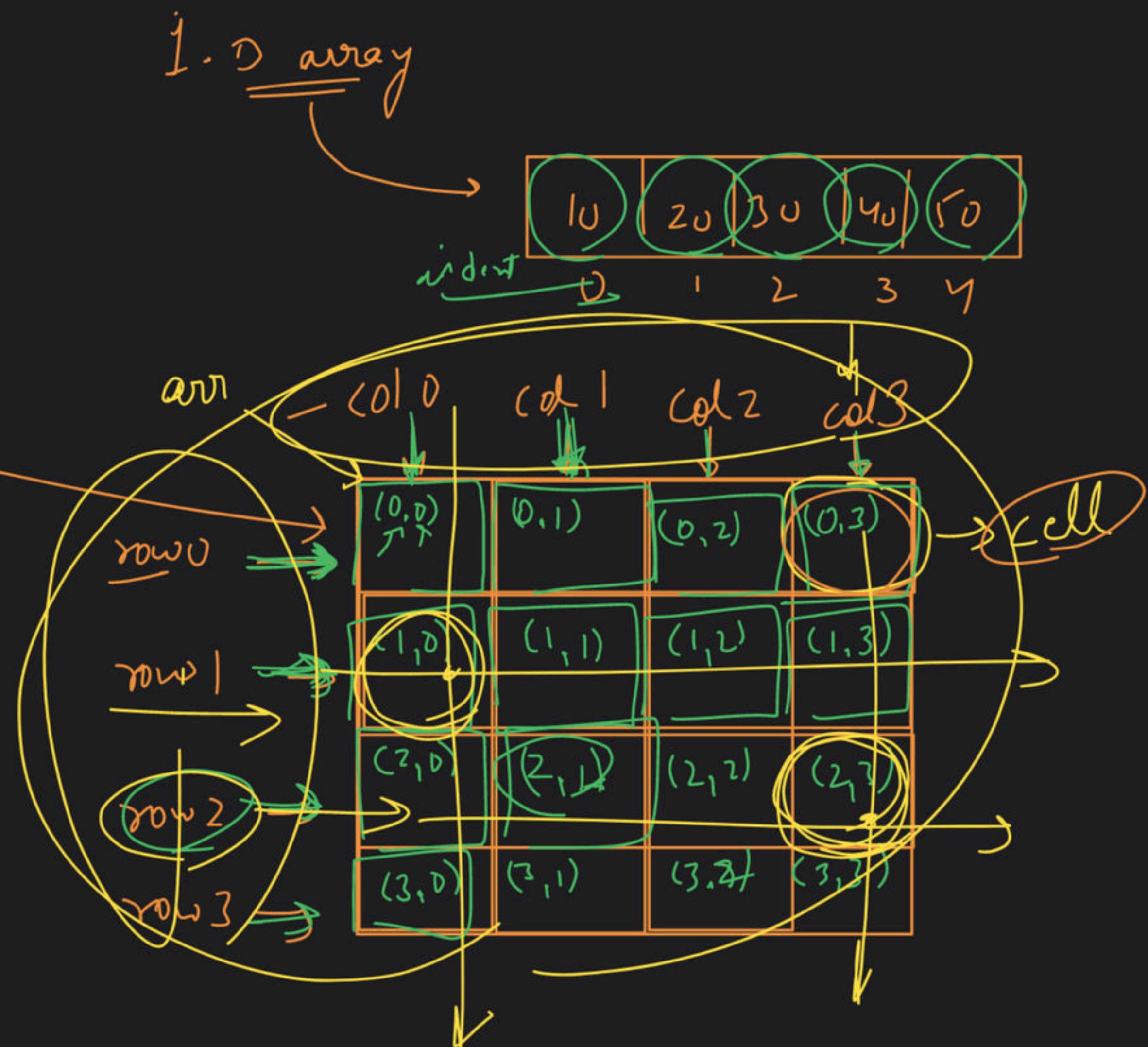
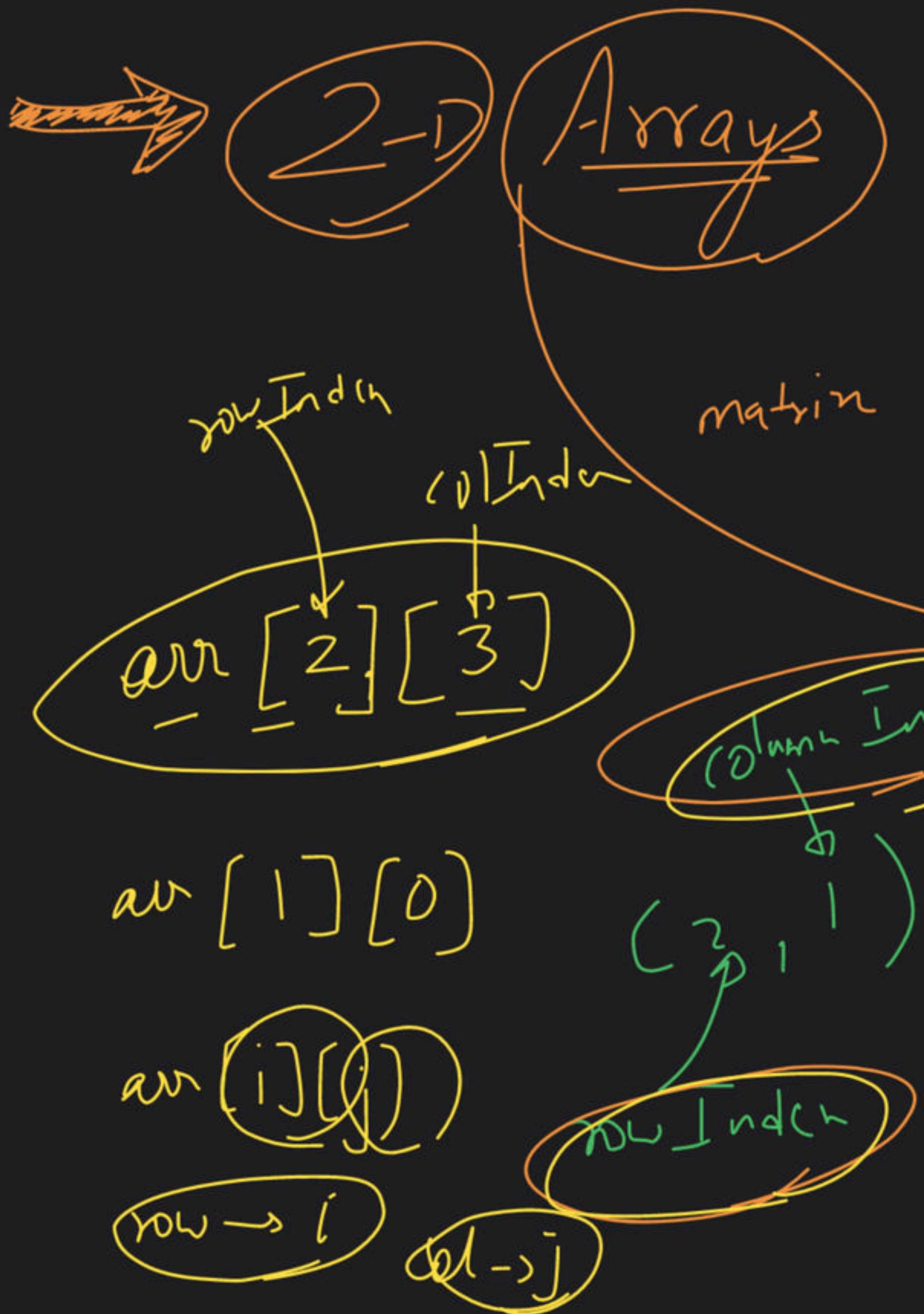
foo(int i=0 ; i < n ; i++)

{ if (arr[i] == target)

return true;

}





formula:-

$$C * I + J$$

$$4 \times 2 + 2$$

$$= 8 + 2 \rightarrow 10$$

$$\text{ans}[2][2] = 35$$

memory

$C \rightarrow$ total no of columns

$$C = 4$$

$i \rightarrow$ row Index

$$i = 2$$

$j = 10$ column Index

$$j = 2$$

2D

0,0	0,1	0,2	0,3
1,0	1,1	1,2	1,3
2,0	2,1	2,2	2,3
3,0	3,1	3,2	3,3

newIndex

(2, 2)

(0, 1)

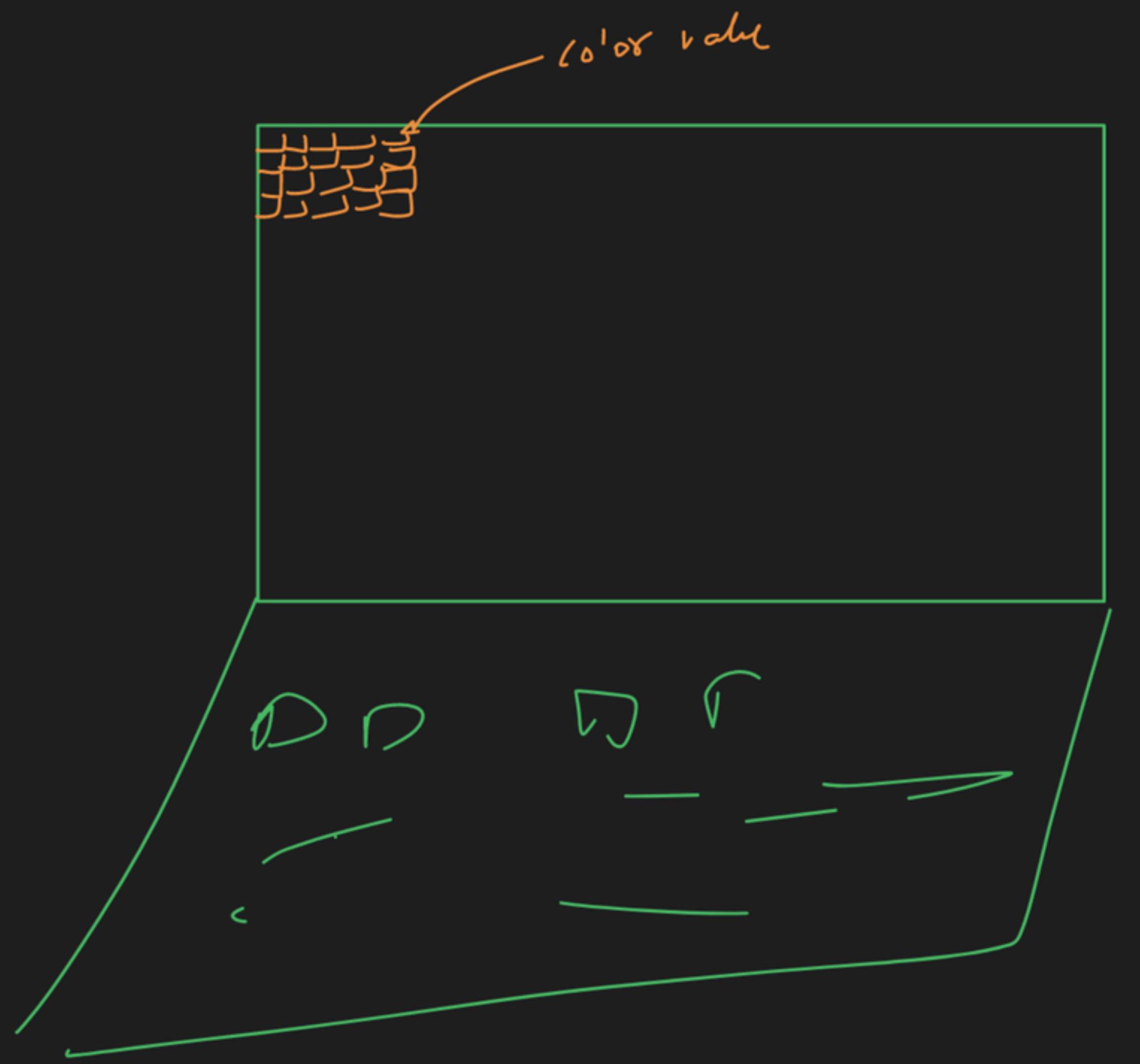
Index



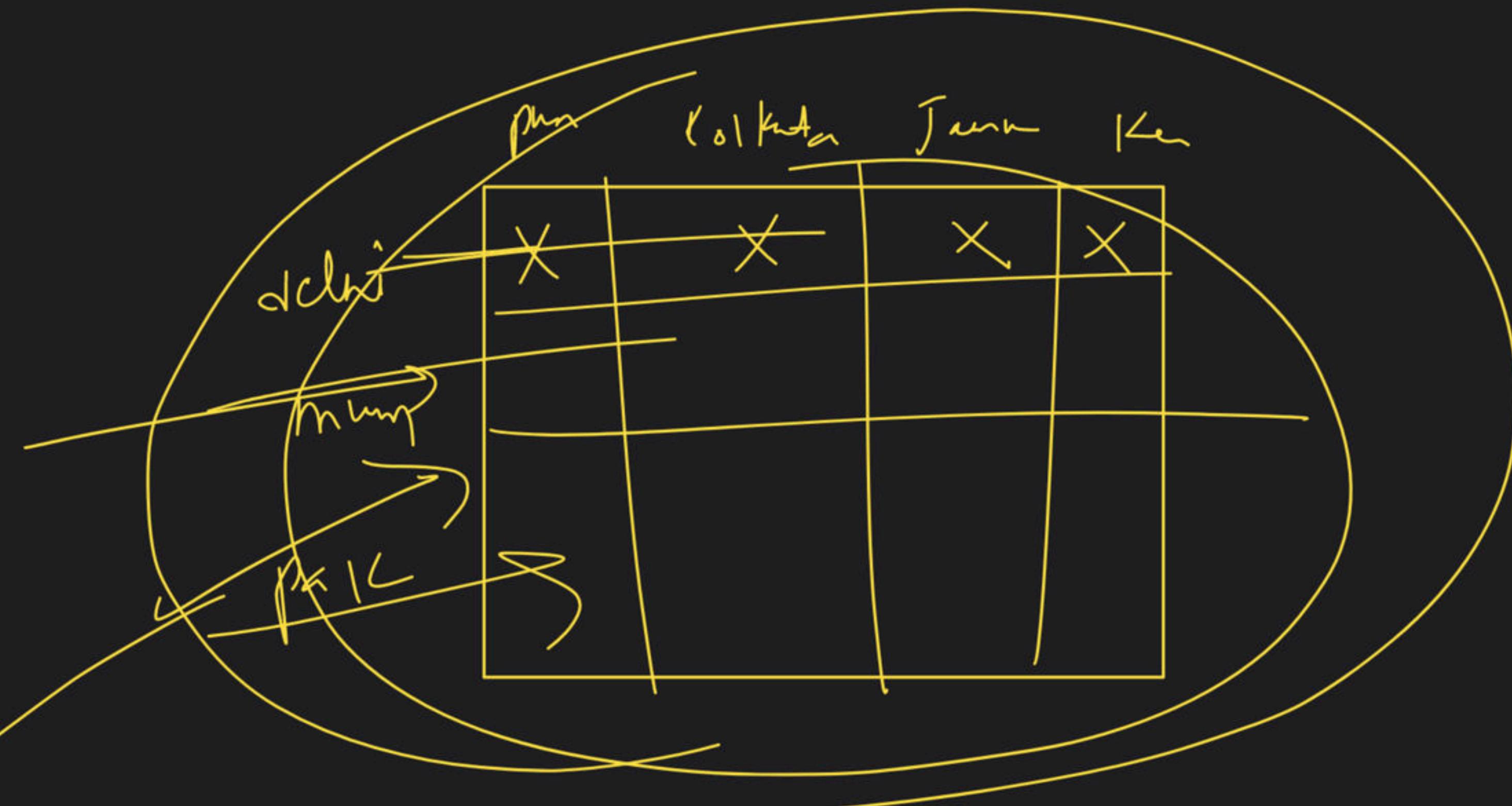
$\text{arr}[\emptyset] \rightarrow \mathbb{S}^-$

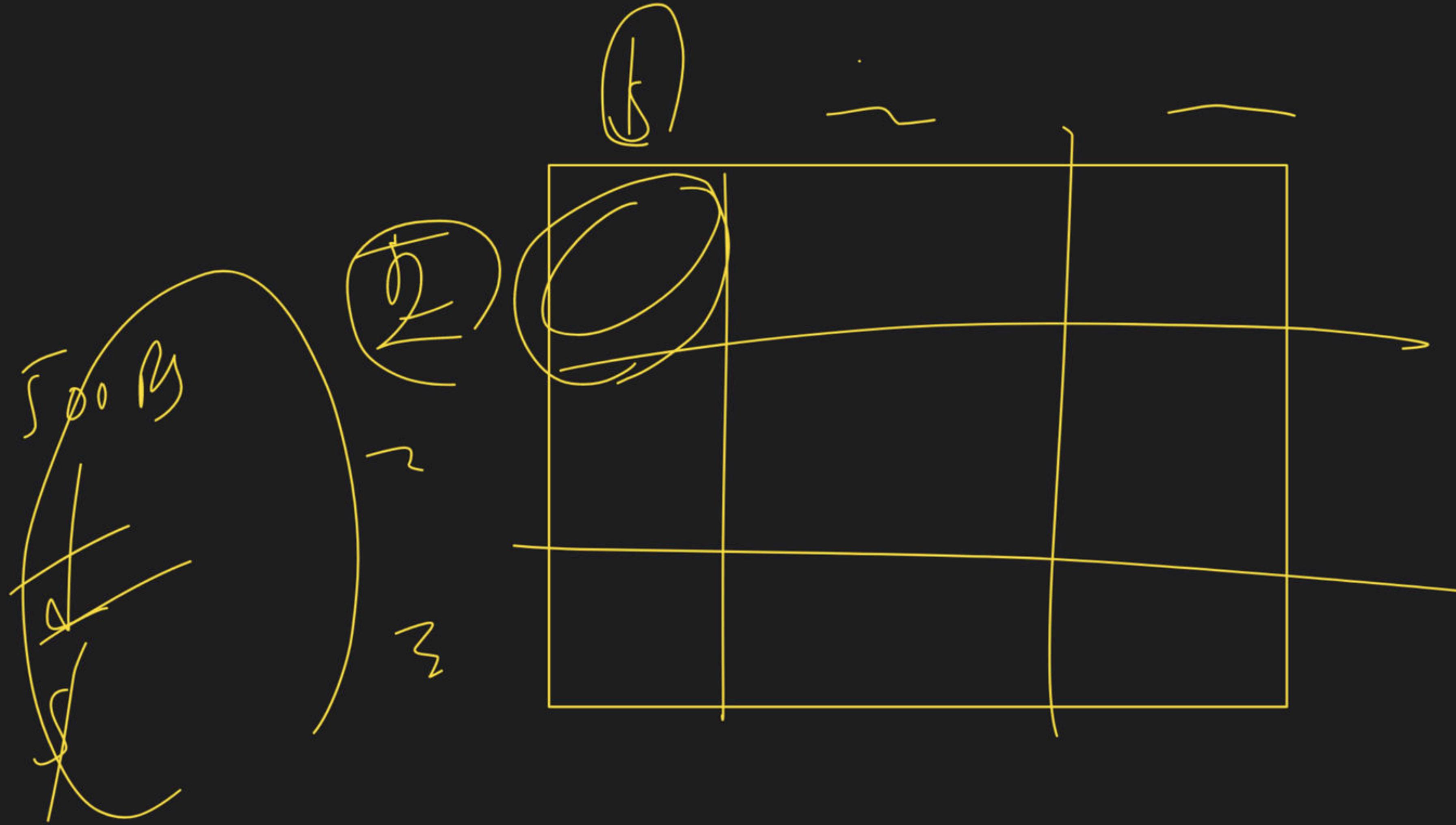
α

$\text{arr}\gamma = \mathbb{J}^-$



Creation:-





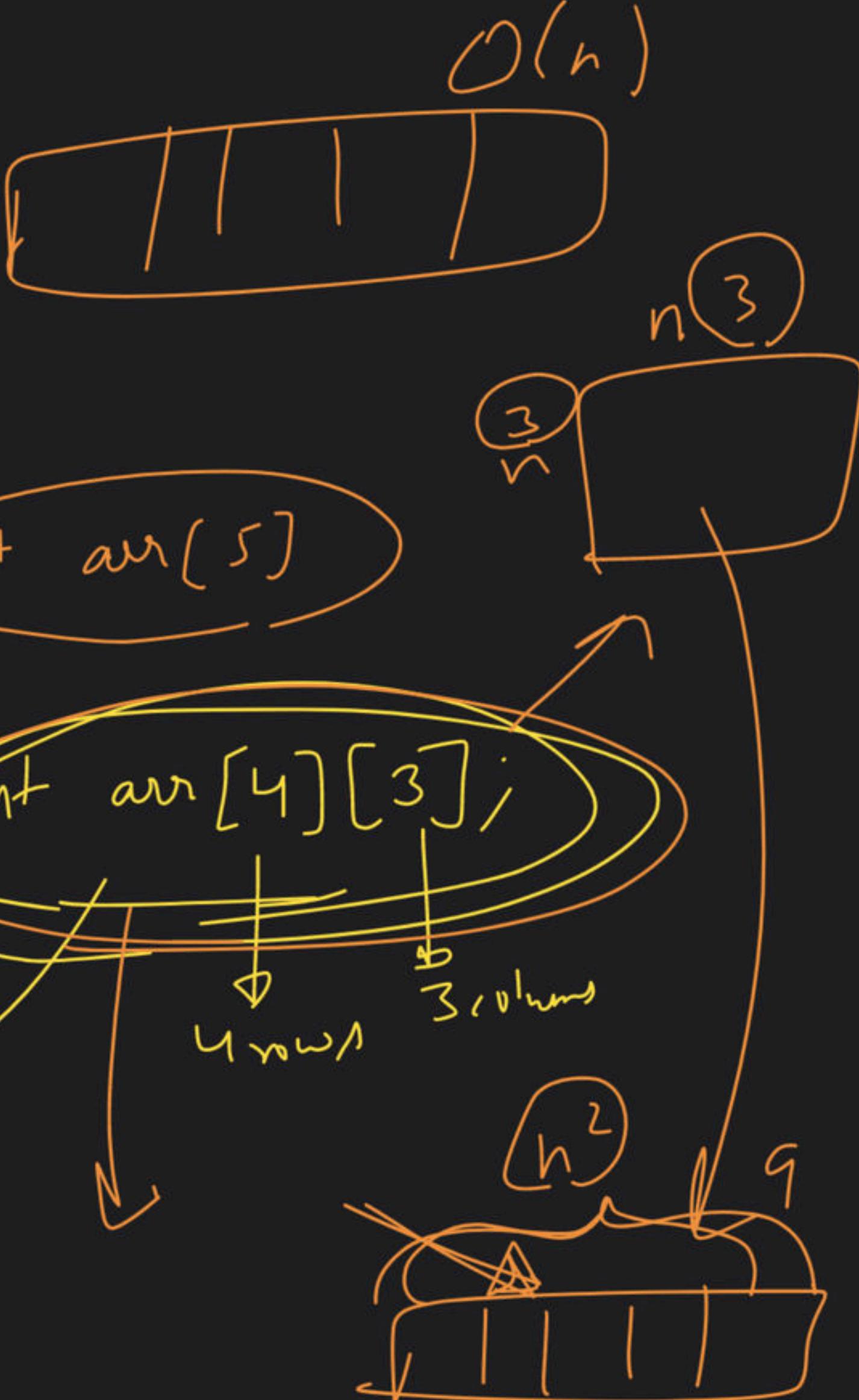
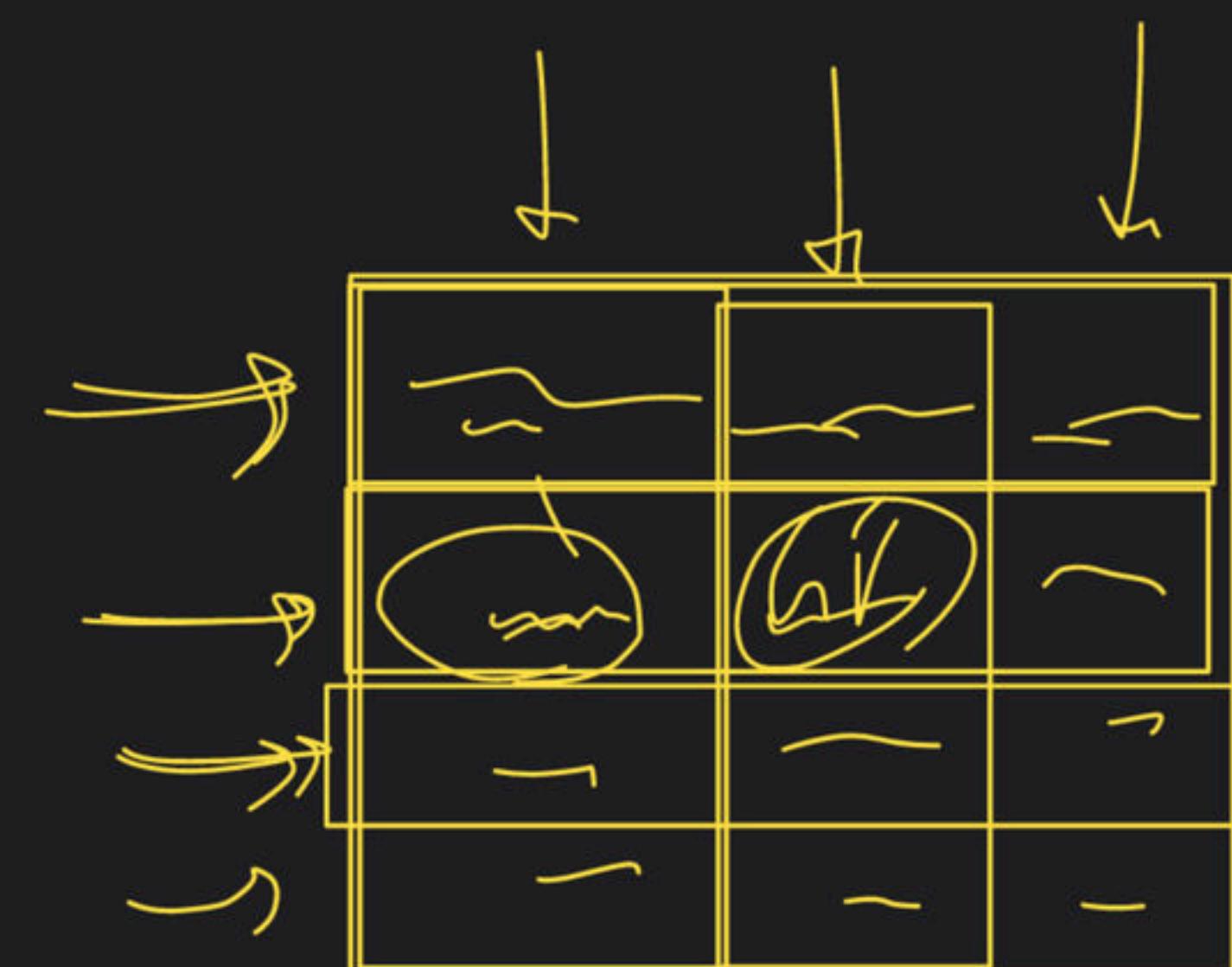
Creation

declare

1D → int arr[5]

2D →

int arr[4][3];

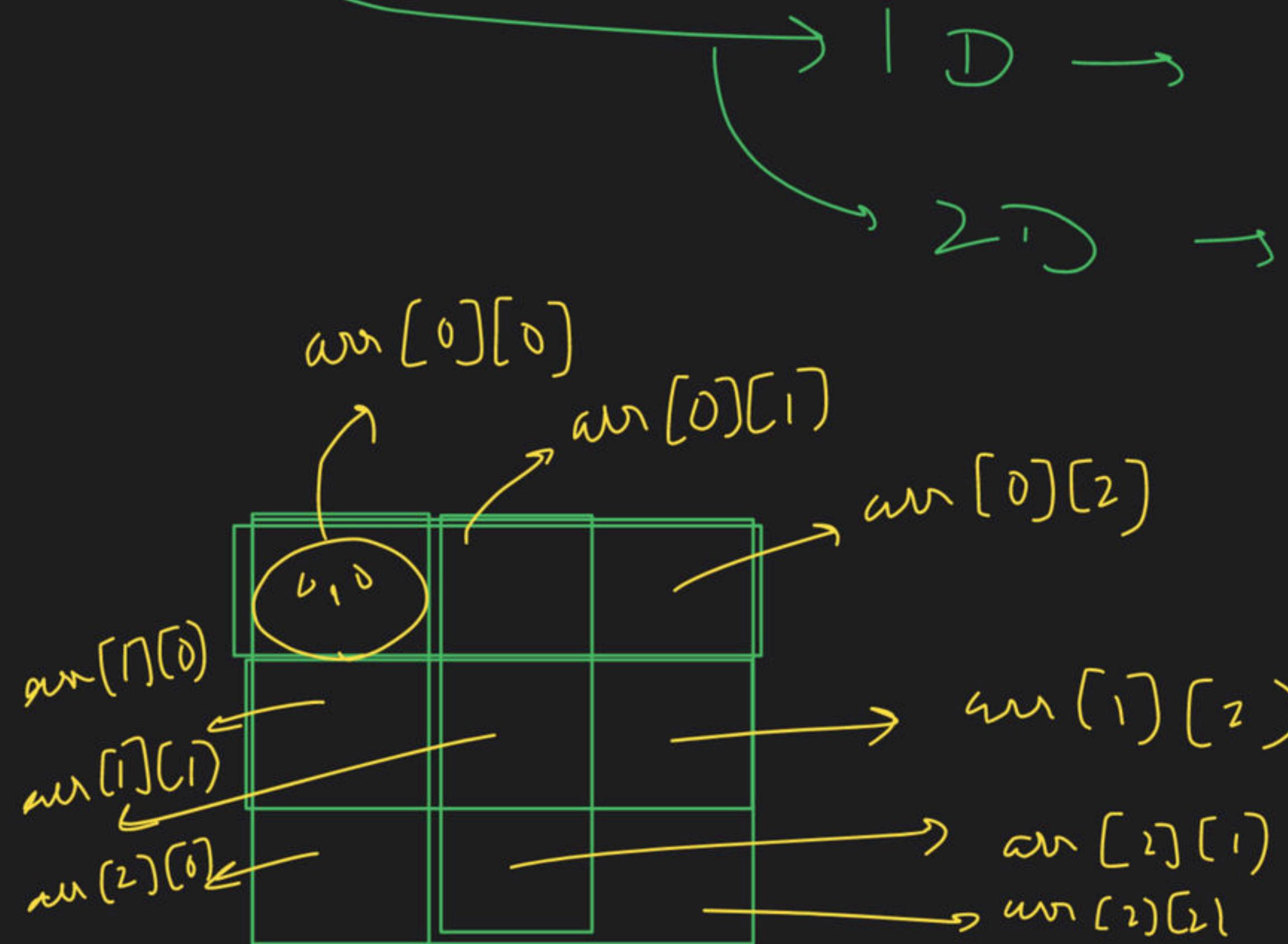


Initialisatio

int arr[4][2] = {10, 20}

int arr[4][3] = { {10, 20, 30},
 {11, 12, 13},
 {15, 16, 17},
 {20, 21, 22} };

Access



| \rightarrow $\text{arr}[i]$

\rightarrow $\text{arr}[i][j]$

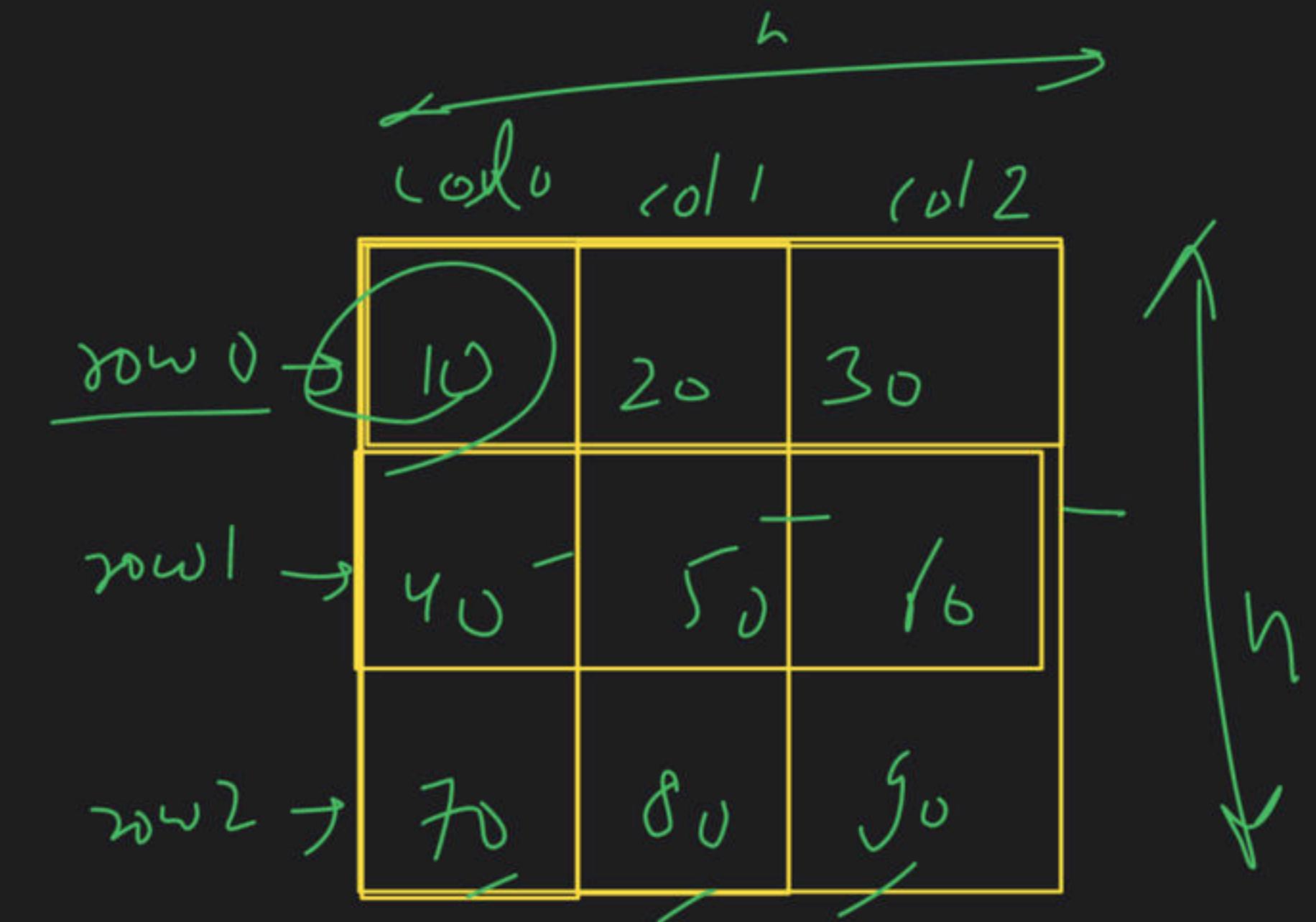
Column Index

RowIndex

```

for {
    for (row < 3 → row < n)
        for (col < 3 → col < n)
            out <- arr[row][col]
}

```



$\text{row} 0 \rightarrow \text{col} 0 \text{ col} 1 \text{ col} 2$ $O(n^2)$
 $\text{row} 1 \rightarrow \text{col} 0 \text{ col} 1 \text{ col} 2$
 $\text{row} 2 \rightarrow \text{col} 0 \text{ col} 1 \text{ col} 2$

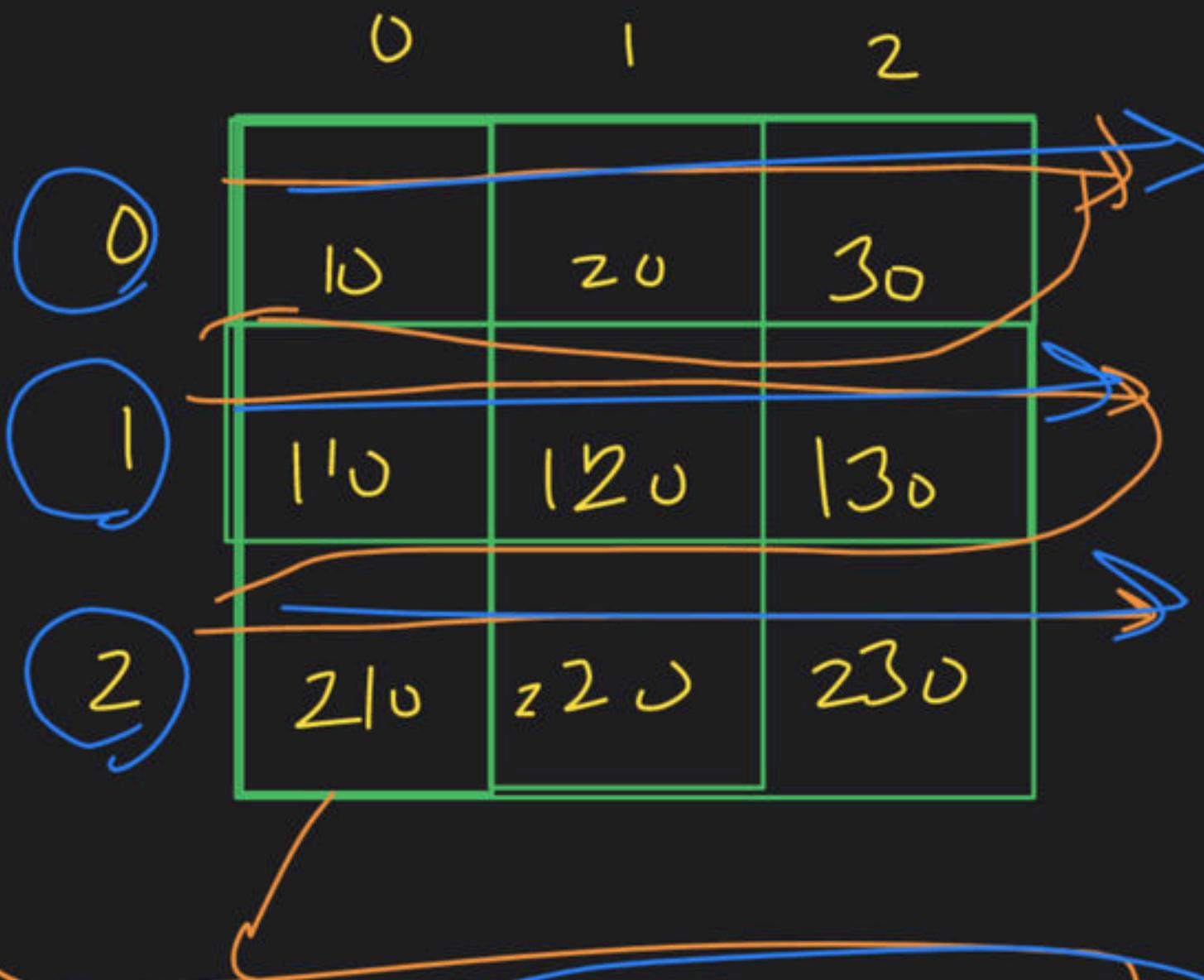
T.C

Fill → 2D array



?

row-wise

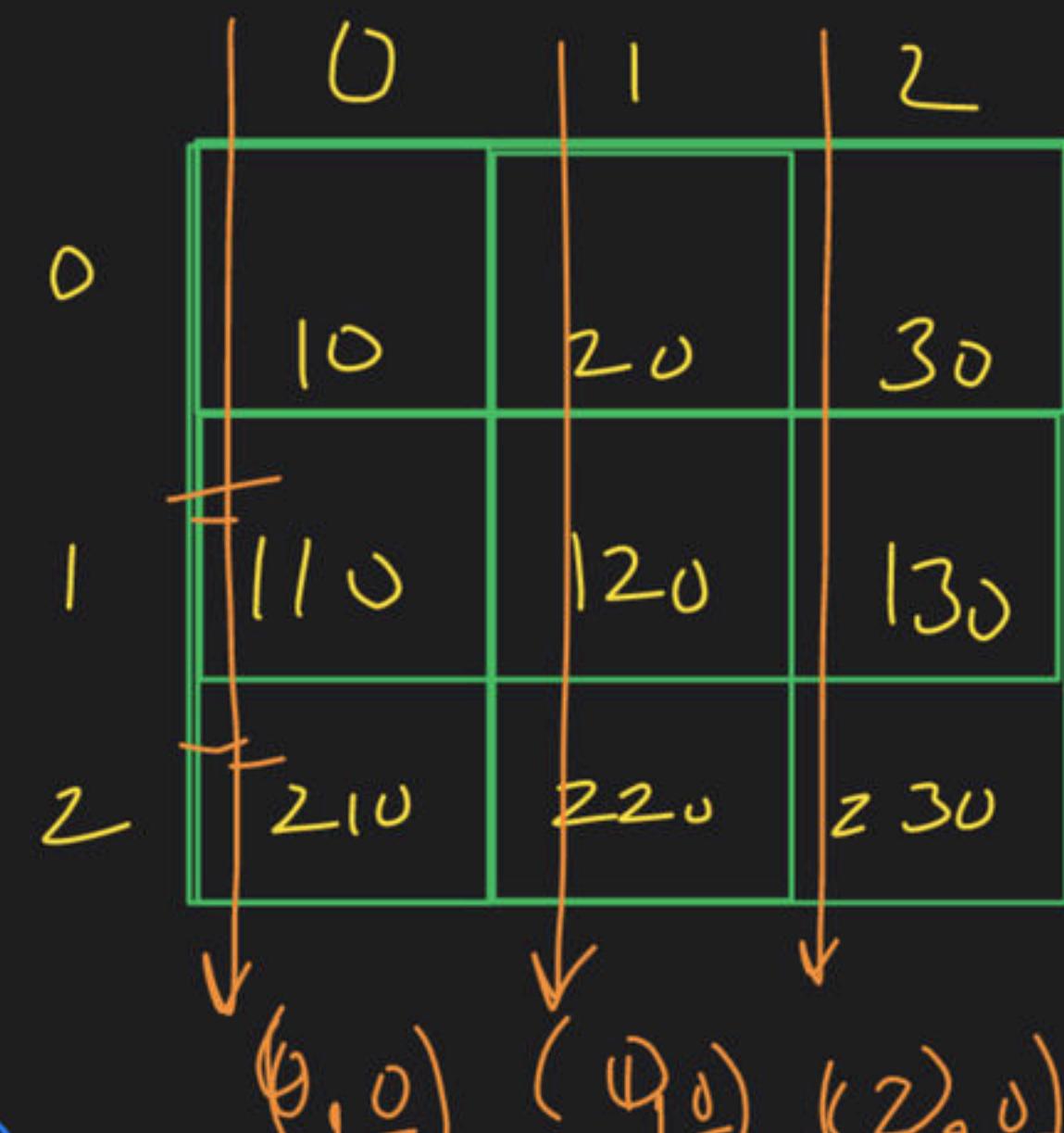


for (row = 0; row < m;)

{
 for (col = 0; col < n;)

{
 arr [row] [col]

column-wise

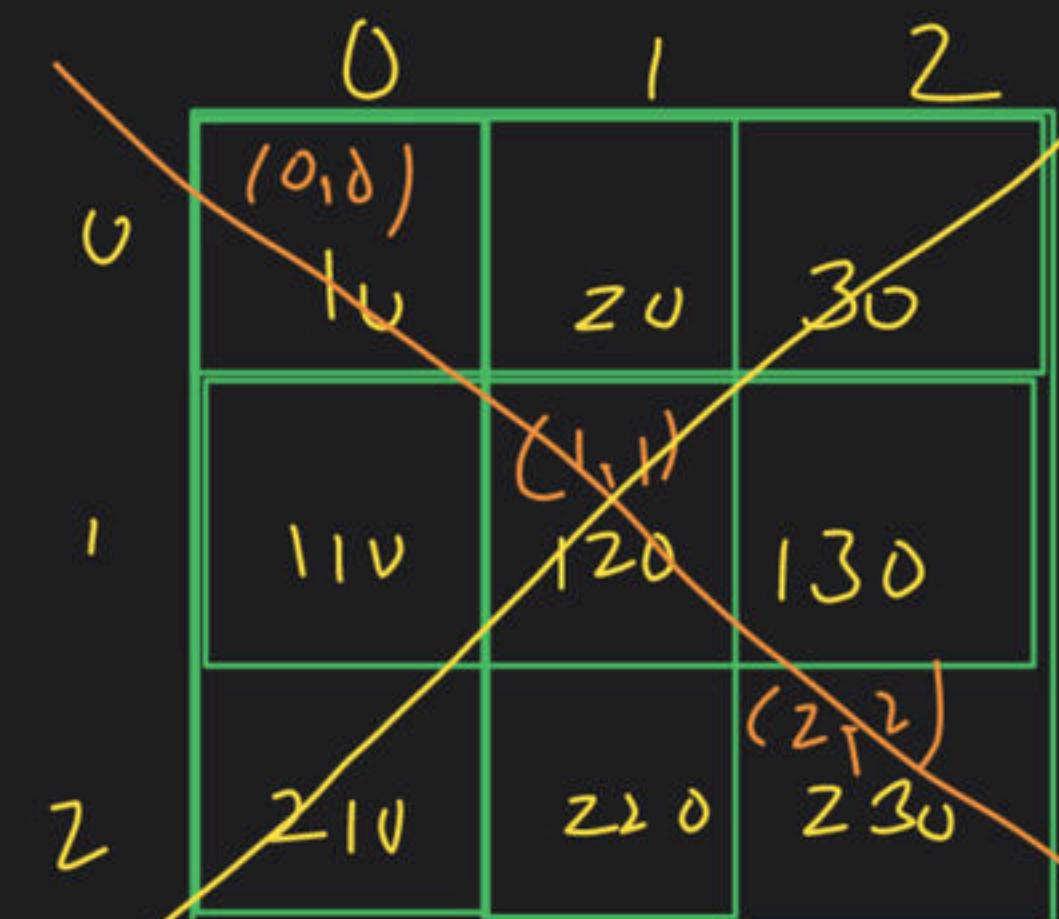


for (row = 0; row < m;)

{
 for (col = 0; col < n;)

{
 arr [row] [col]

diagonal traversal



for (row = 0; row < m;)

{
 for (col = 0; col < n;)

{
 if (row == col)

{
 cout << arr[row][col]

```

for (r=0 → < 3)
{
    for (c=0 → < 2)
    {
        arr [c][r]
    }
    cout << endl;
}

```

$r = 0$
 $c = 0$
 $arr[0][0]$
 $c = 1$
 $arr[1][0]$

arr

10	20
30	40
50	60

$r = 0$
 $r = 1$
 $r = 2$
 $c = 0$
 $c = 1$

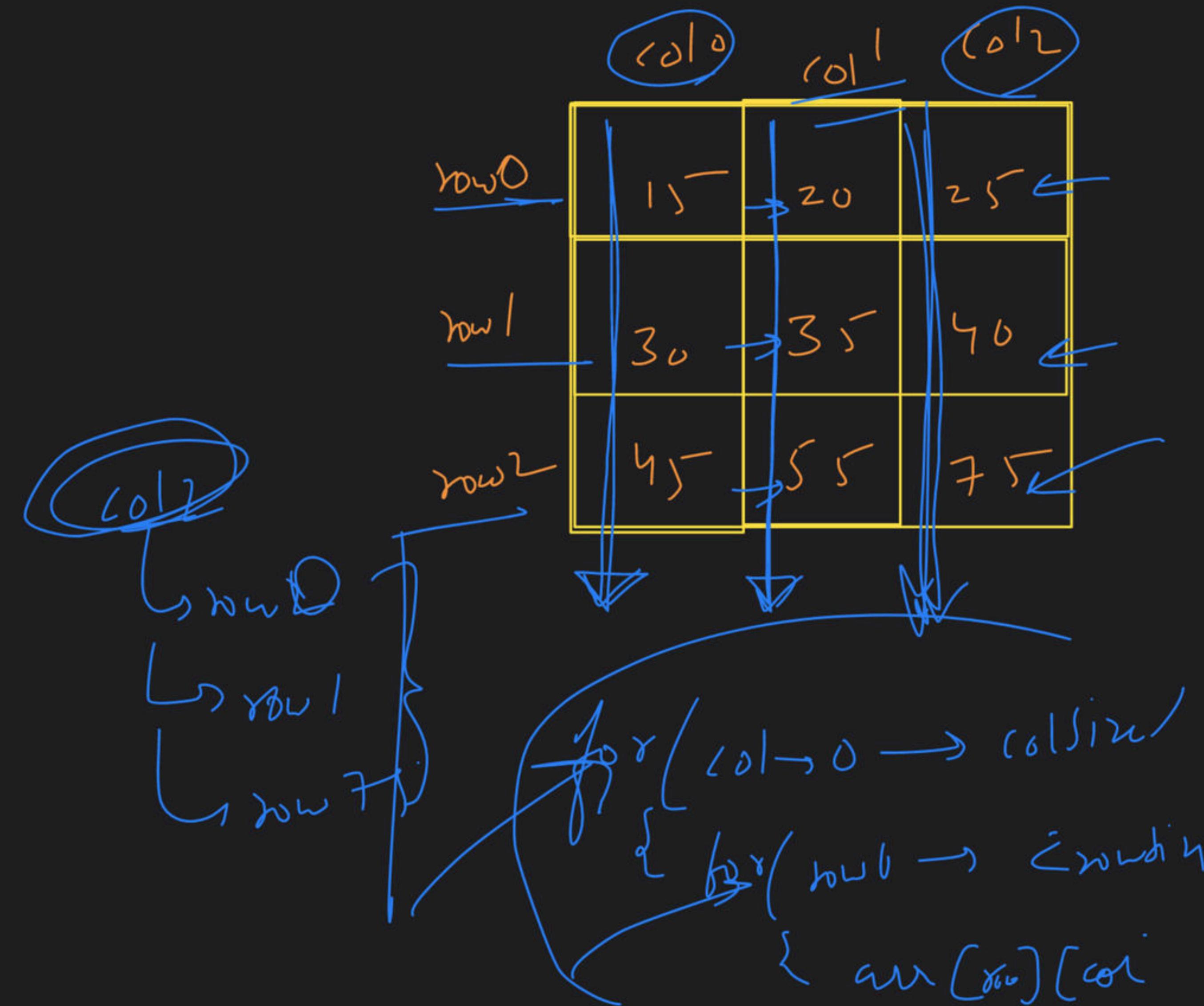
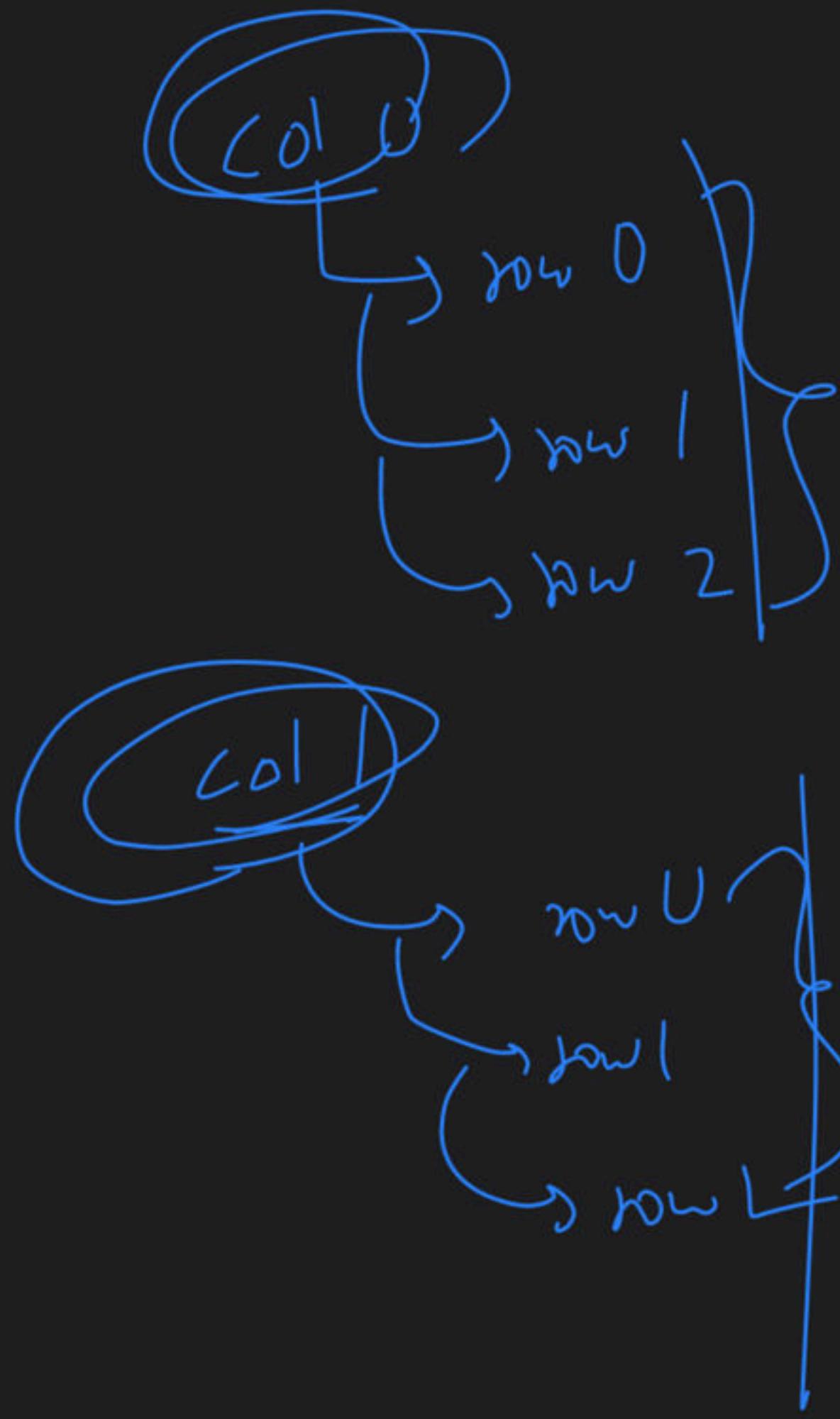
Sahan matrix
 $\text{arr}[c][r]$
 $\text{arr}[r][c]$
 $\text{arr}(r)(c)$

$\text{for} (c=0 \rightarrow 2)$
 $\{$
 $\text{for} (r=0 \rightarrow 3)$





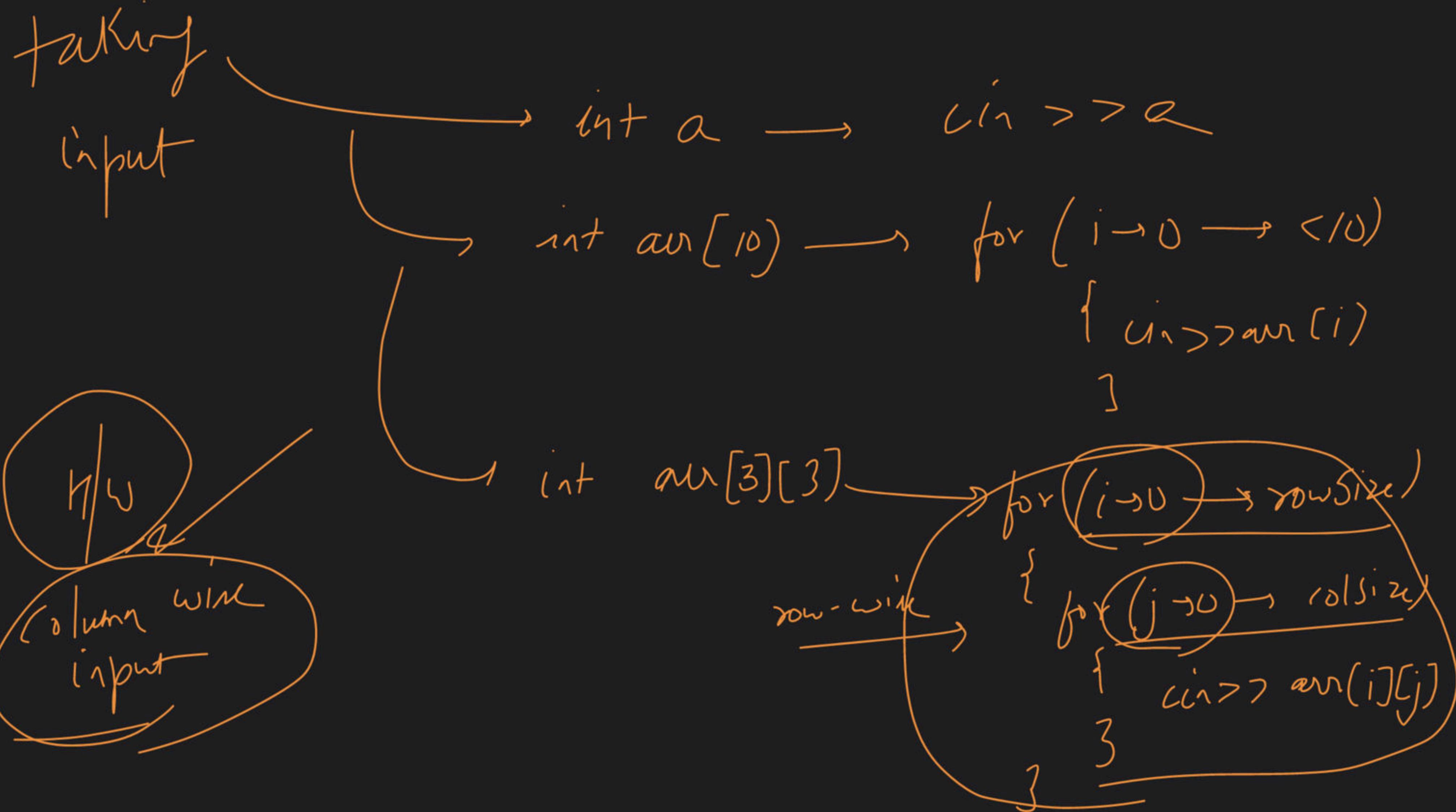




2

min

Break



→ Timi Break → Paan'

← Cushtion

Searching
=

if ($arr[i][j] == \text{target}$)

return true

arr

0	10	20	30	40
1	50	55	57	59
2	69	72	79	89
3				

return true

target → 79

2D array with function

int main()

{ int arr[2][2] = {0}

int rowsize = 2, colsiz = 2

solve(arr, rowsize, colsiz);

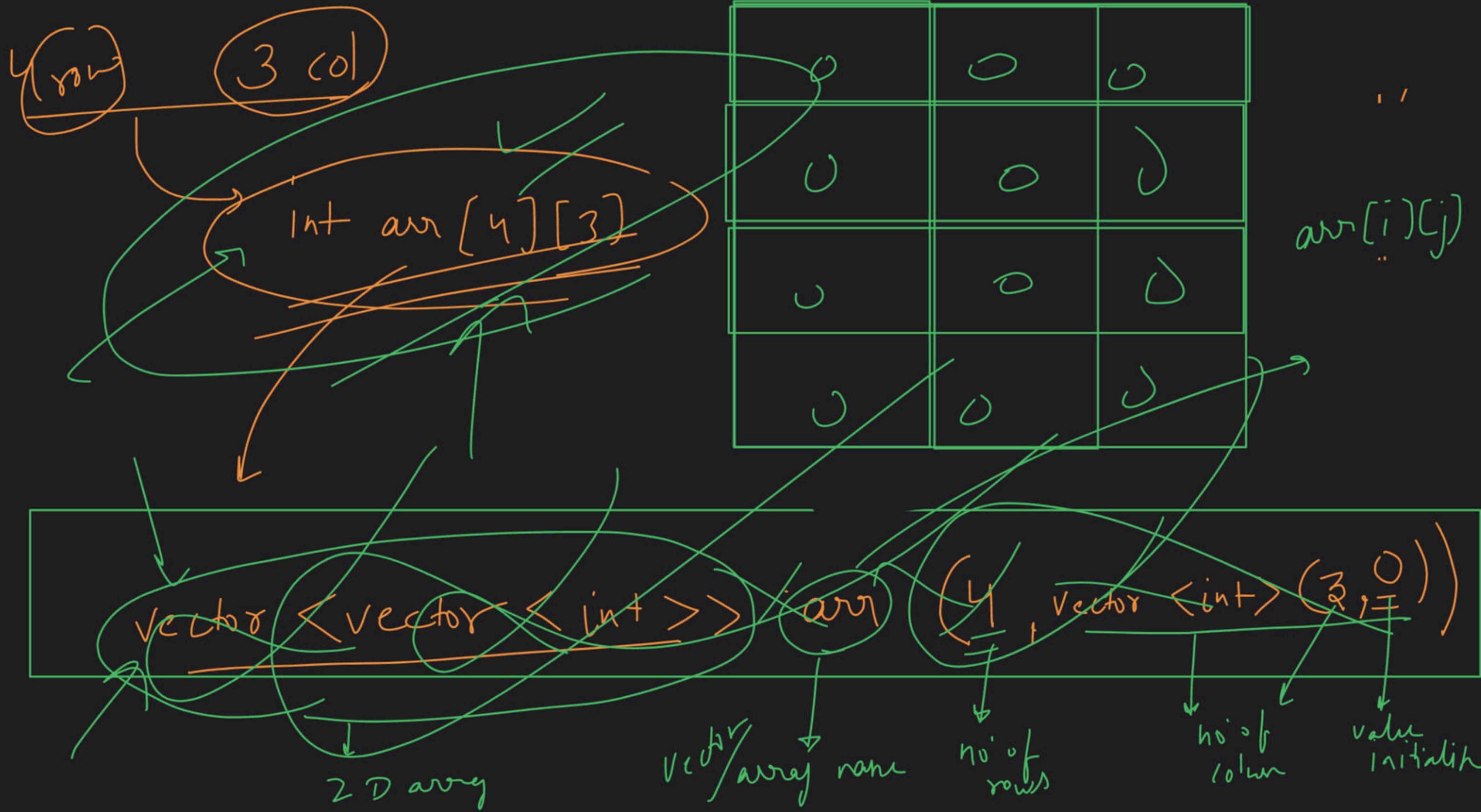
}

}

2D \rightarrow 1D
 $c * i + j$

void solve (int arr[2][2],
int rowsize,
int colsiz)

column



→ find minimum value

int minValue = ~~INT MAX~~

for (i=0 → 85)

{
for (j=0 → <ω)

} minValue = min (minValue,
 arr[i][j])

}

finalAns

0	70	23	41
1	27	69	42
2	56	12	18

① row-wise sum

row 0
int sum = 0
 $10 + 20 + 30 \rightarrow \text{print sum}$

row 1
int sum = 6
 $10 + 11 + 12 \rightarrow \text{print sum}$

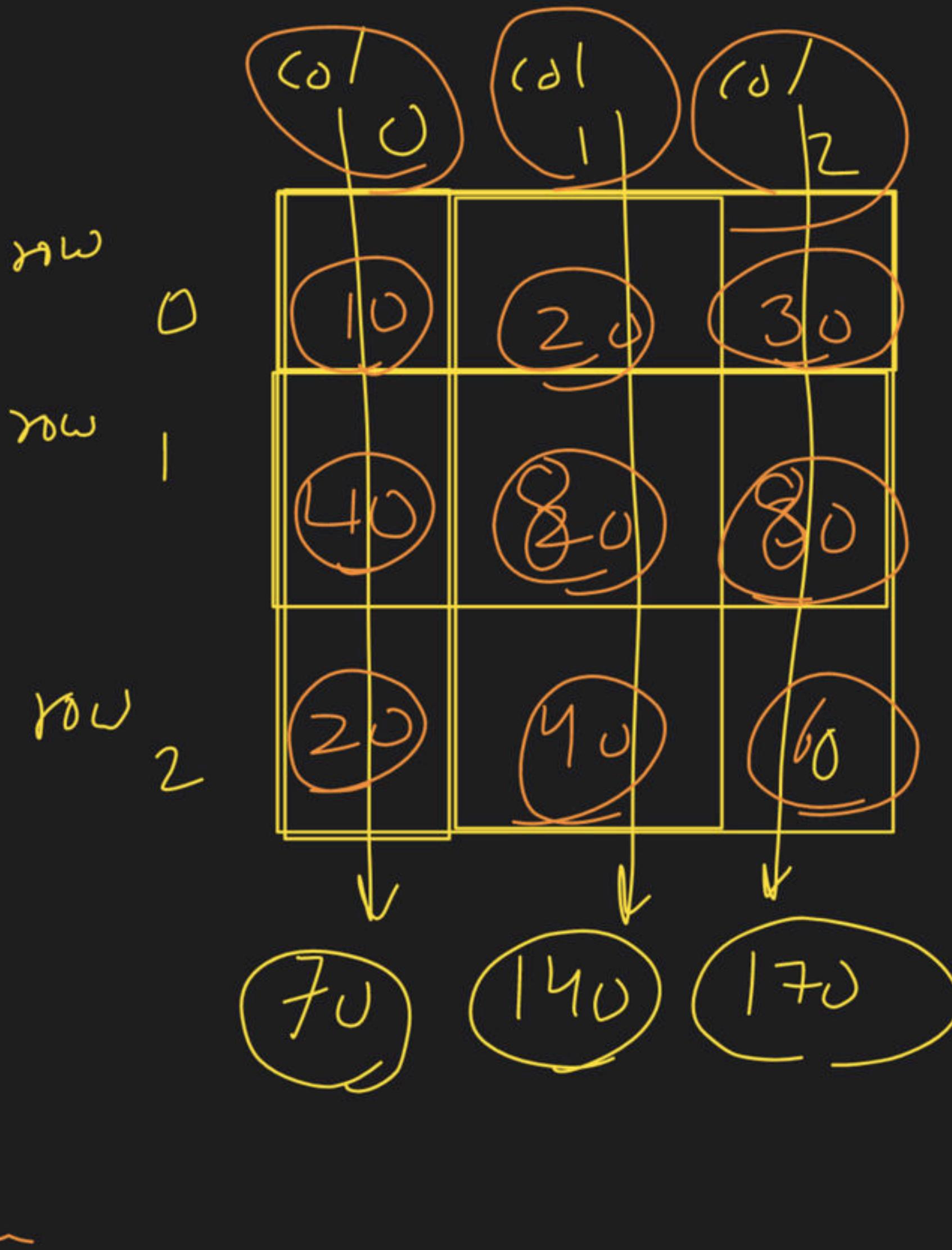
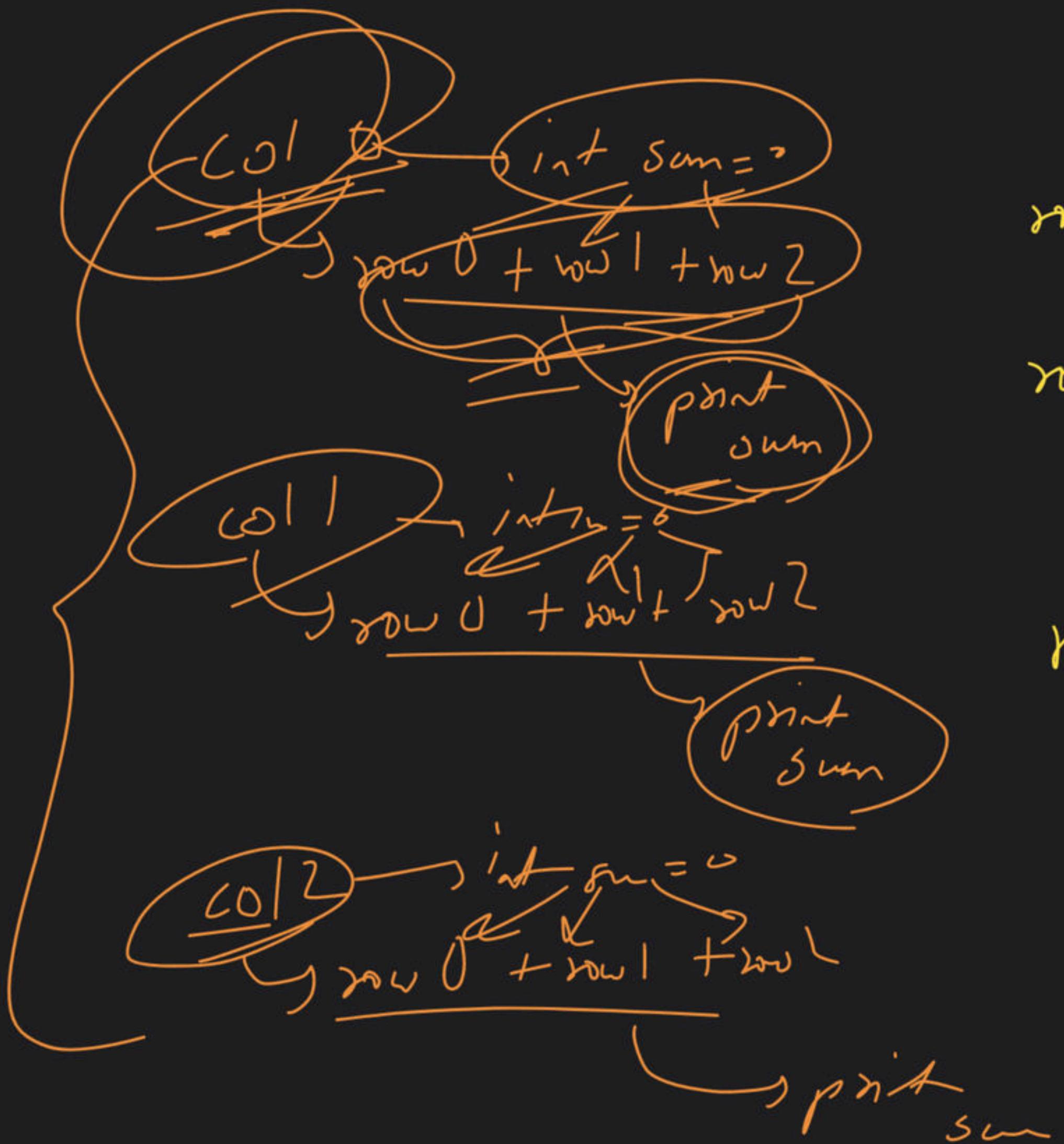
row 2
int sum = 0
 $70 + 80 + 90 \rightarrow \text{print sum}$

Print

0	1	2
10	20	30
10	50	60
70	80	90

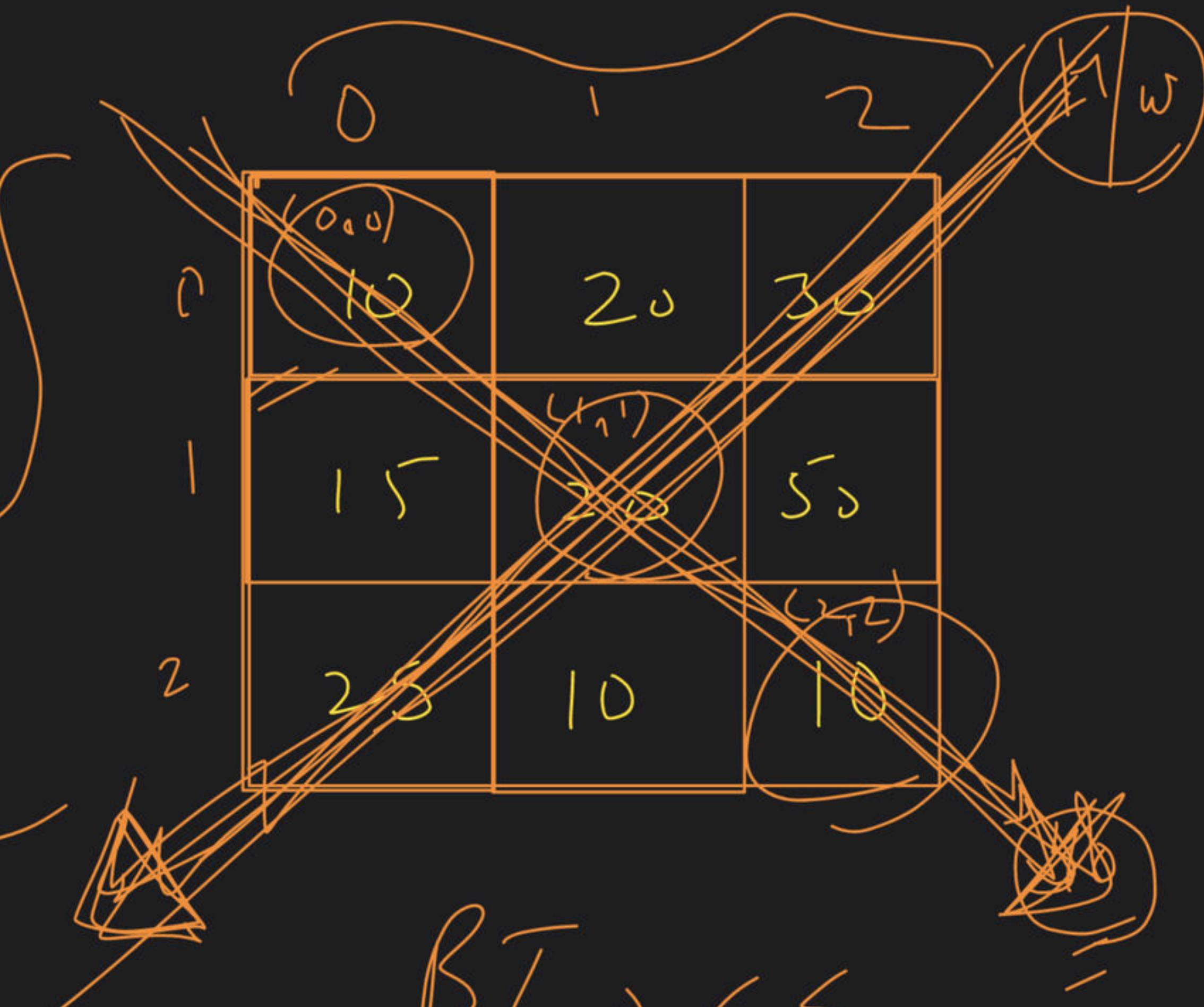
60
150
240

for ($i=0 \rightarrow n$)
for ($j=0 \rightarrow m$)
if sum < point



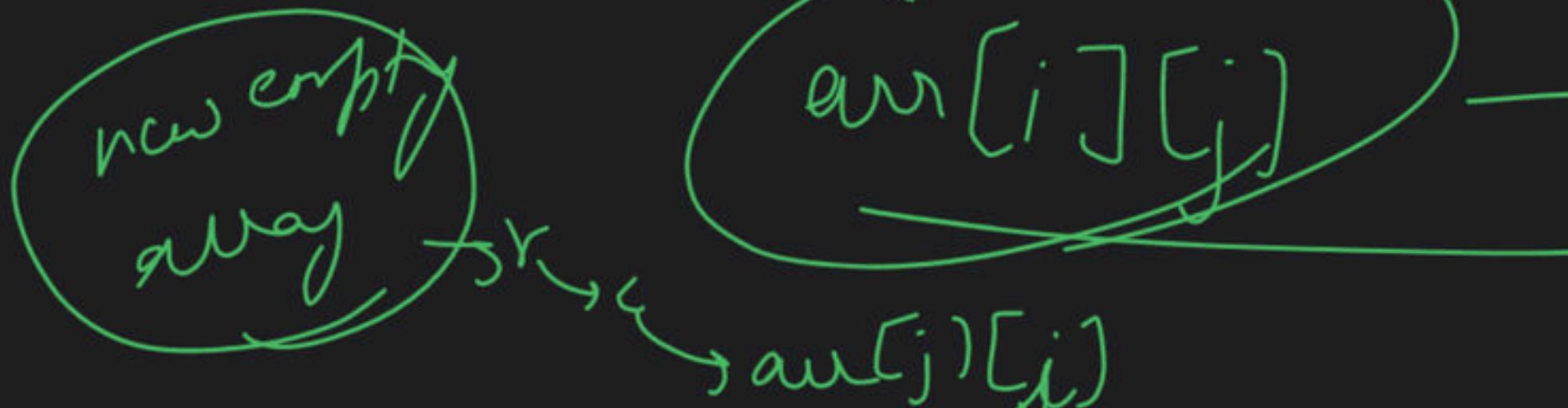
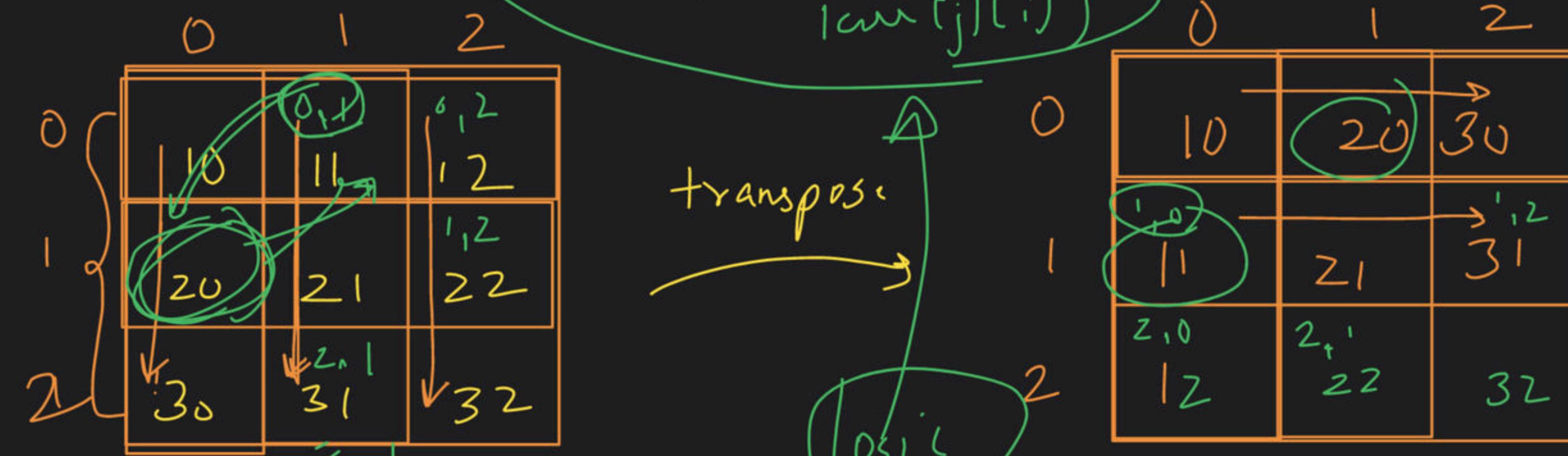
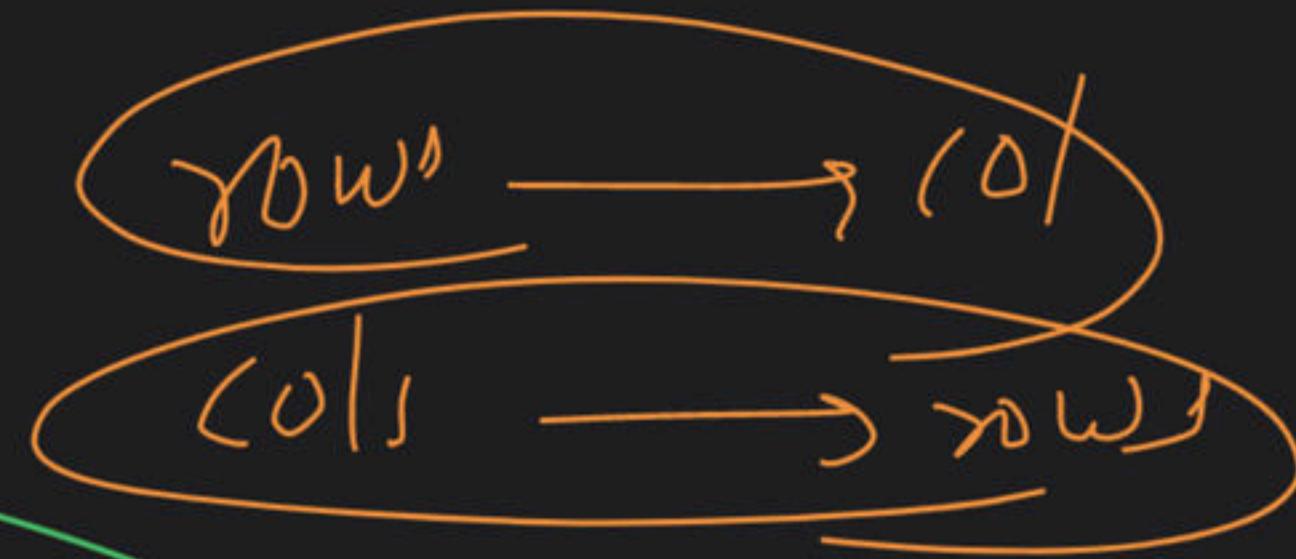
$sum = sum + arr[i][i]$

$0 \rightarrow < 3$

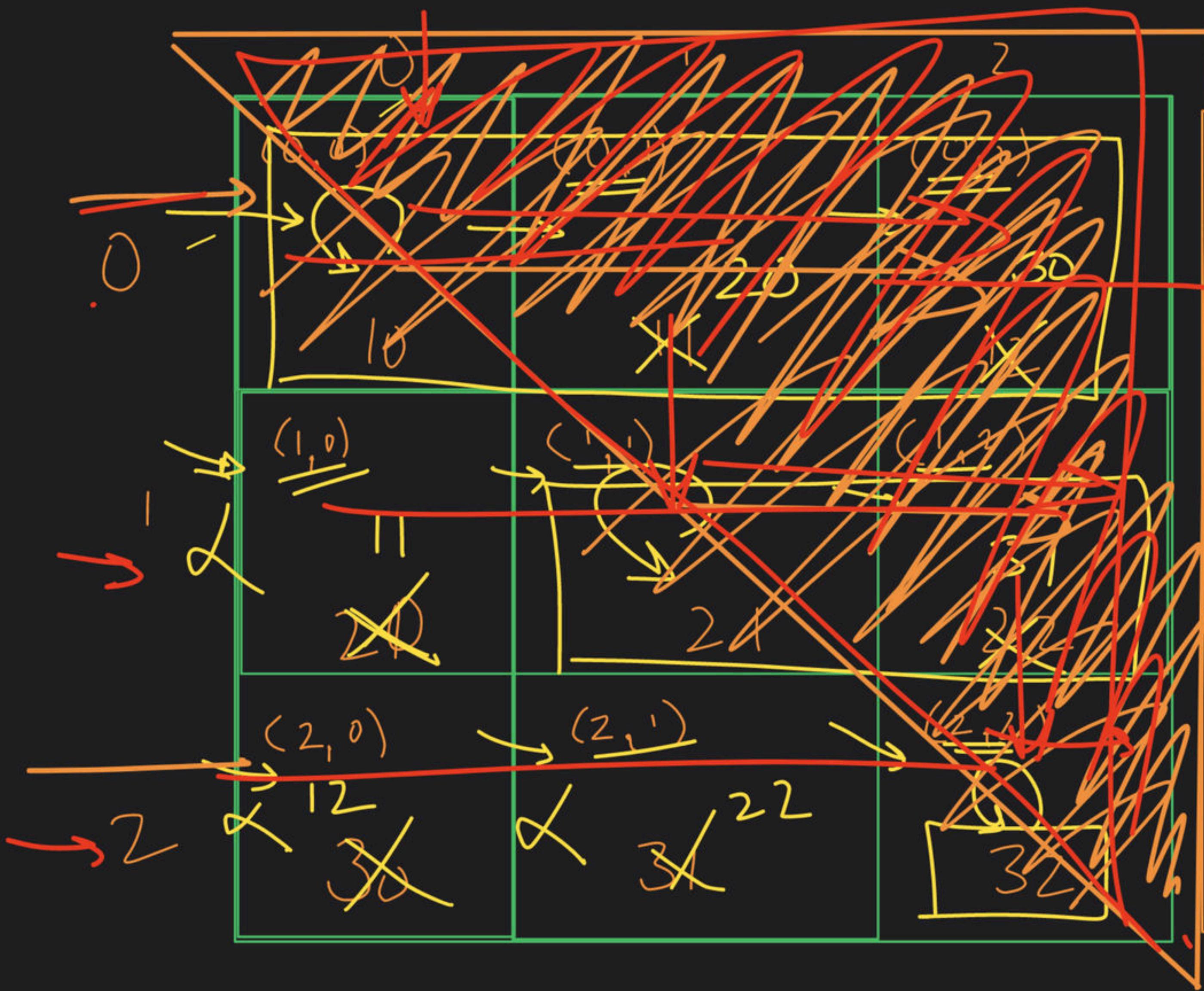


$BT \rightarrow 55$

① Transpose of Matrix



$arr[j][i]$



Sunday

week 3

Mega Class

Arrays

try

assignments



