unacademy

Arrays - Part II

Course on Data Structure

Pankaj Sharma • Lesson 3 • July 10, 2024

unacademy

CS & IT Engineering

Data Structure

Arrays- III

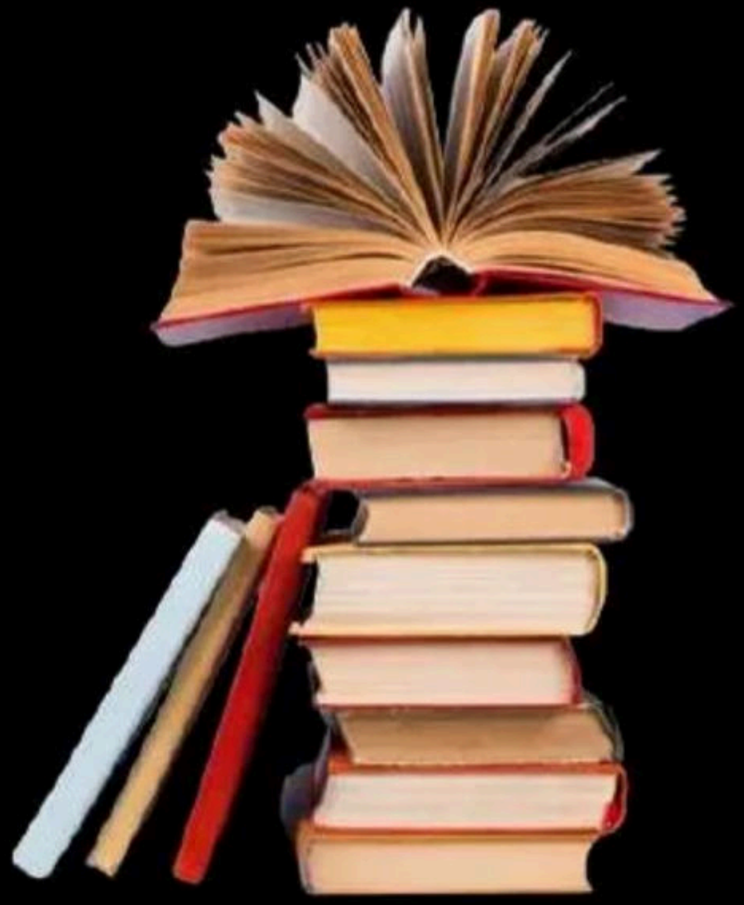
Lecture Number- 03

By- Pankaj Sir

unacademy

Topics

to be covered



1. Arrays

(1) RMO

$A[-5..5][-3..3]$ — $5-(-5)+1$, $3-(-3)+1$

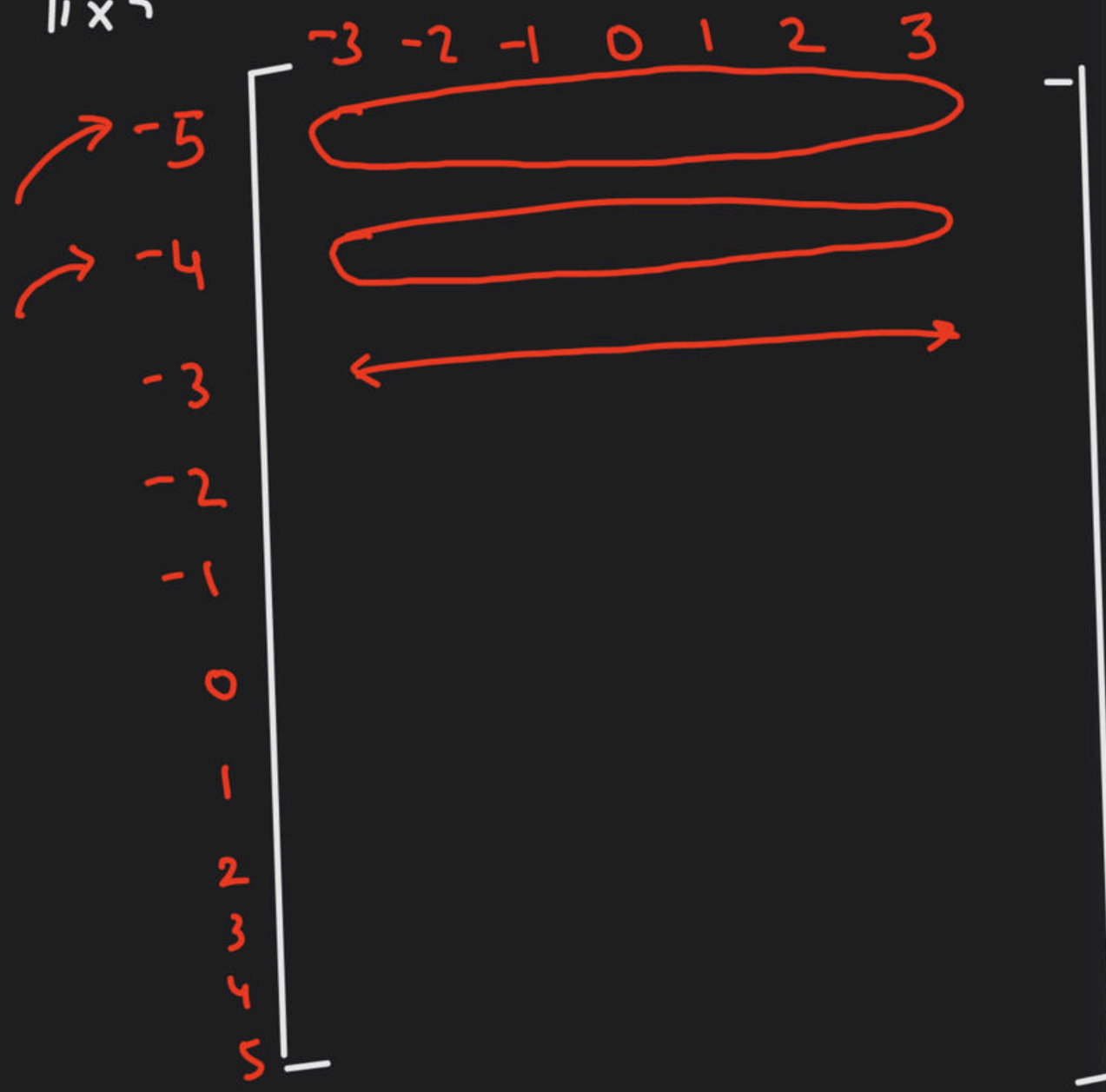
11 × 7

w = 2 bytes, BA = 1000

address (A[1][1])

$A[-5..5][-3..3]$

Yahan ka Har No ⇒ Ek Ele



unacademy

$A[-5..5][-3..3]$

A_{11}

Yahan ka Har No ⇒ 7ele

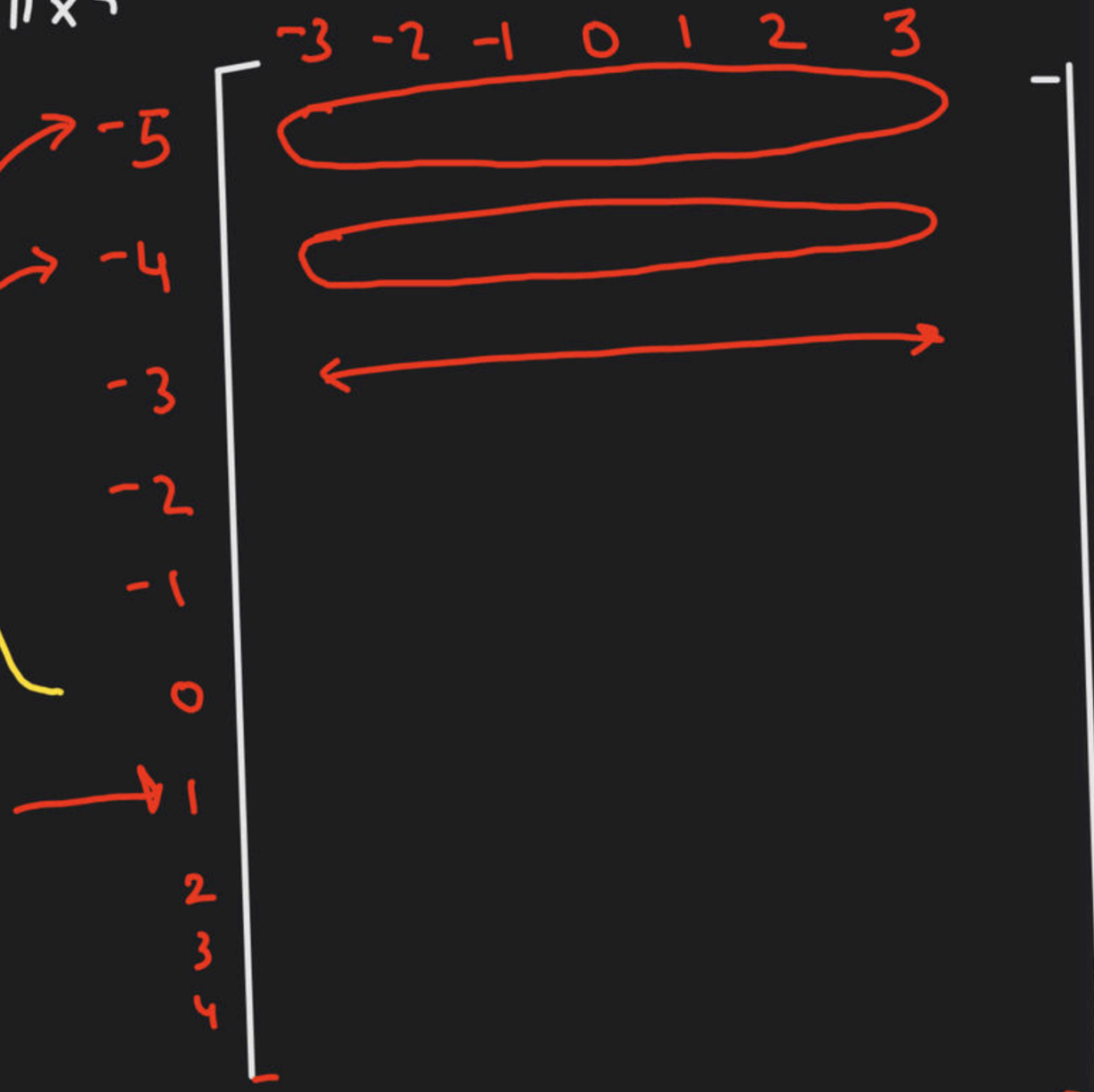
① Rows already filled

= index -5 to 0

= 0 -(-5)+1 = 6 rows

with index -5, -4, -3, -2, -1, 0

Rows already filled



unacademy

$A[-5..5][-3..3]$

A_{11}

11 x 7

Addr(A[1][1])

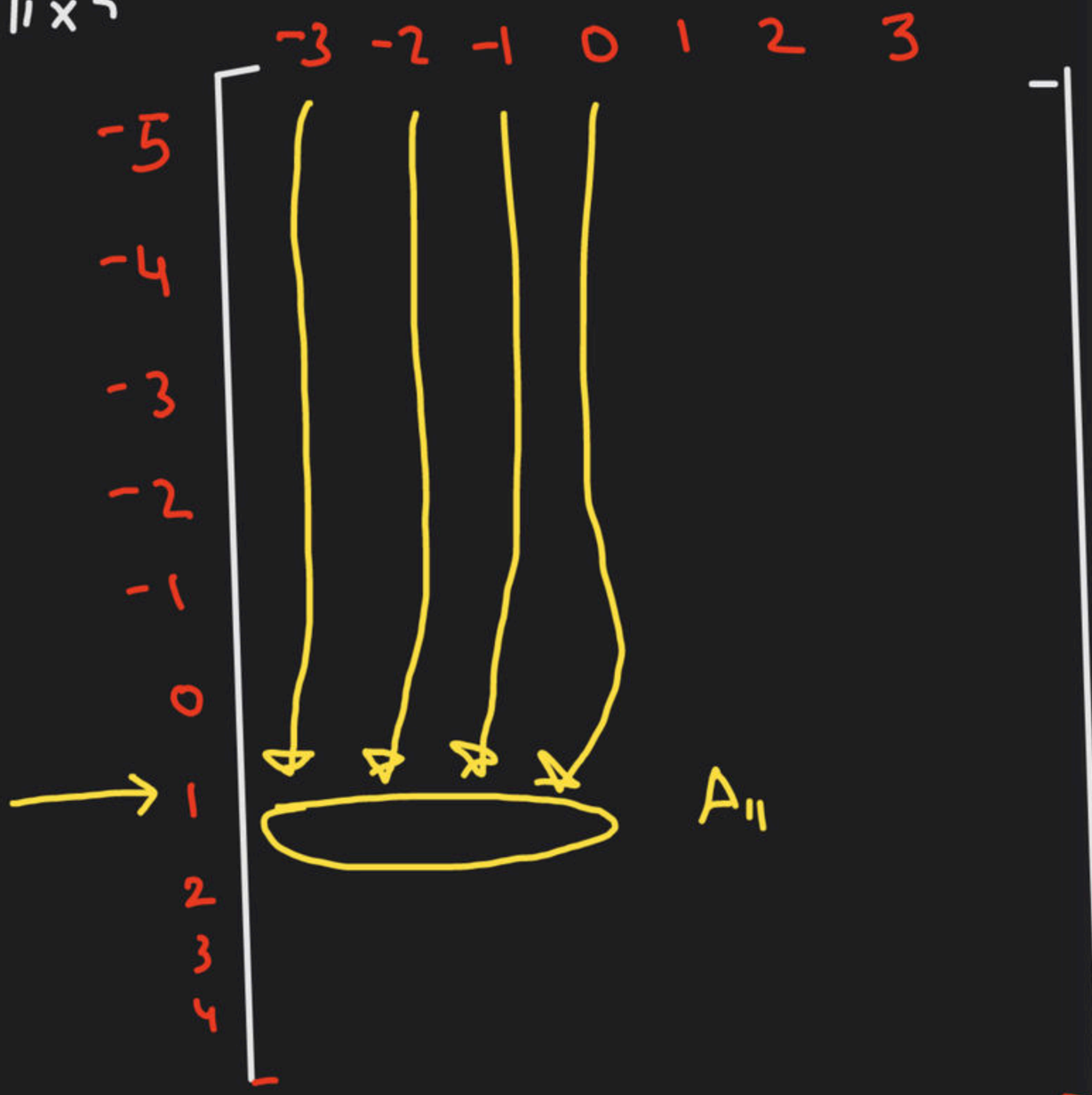
after 5 rows

within row with index 1,

Elem already filled

= Col index -3 to 0

= 0 - (-3) + 1 = 4 elements



6 rows & 4 elements A_{11} is stored

Elements already filled before A[1][1]

$= 6 \times 7 + 4 = 46$ elements

Memory already filled before $A_{11} = 46 \times 2 = 92$ Bytes

addr(A_{11})

$= BA + 92$

$= 1000 + 92$

$= 1092$

2. RMO

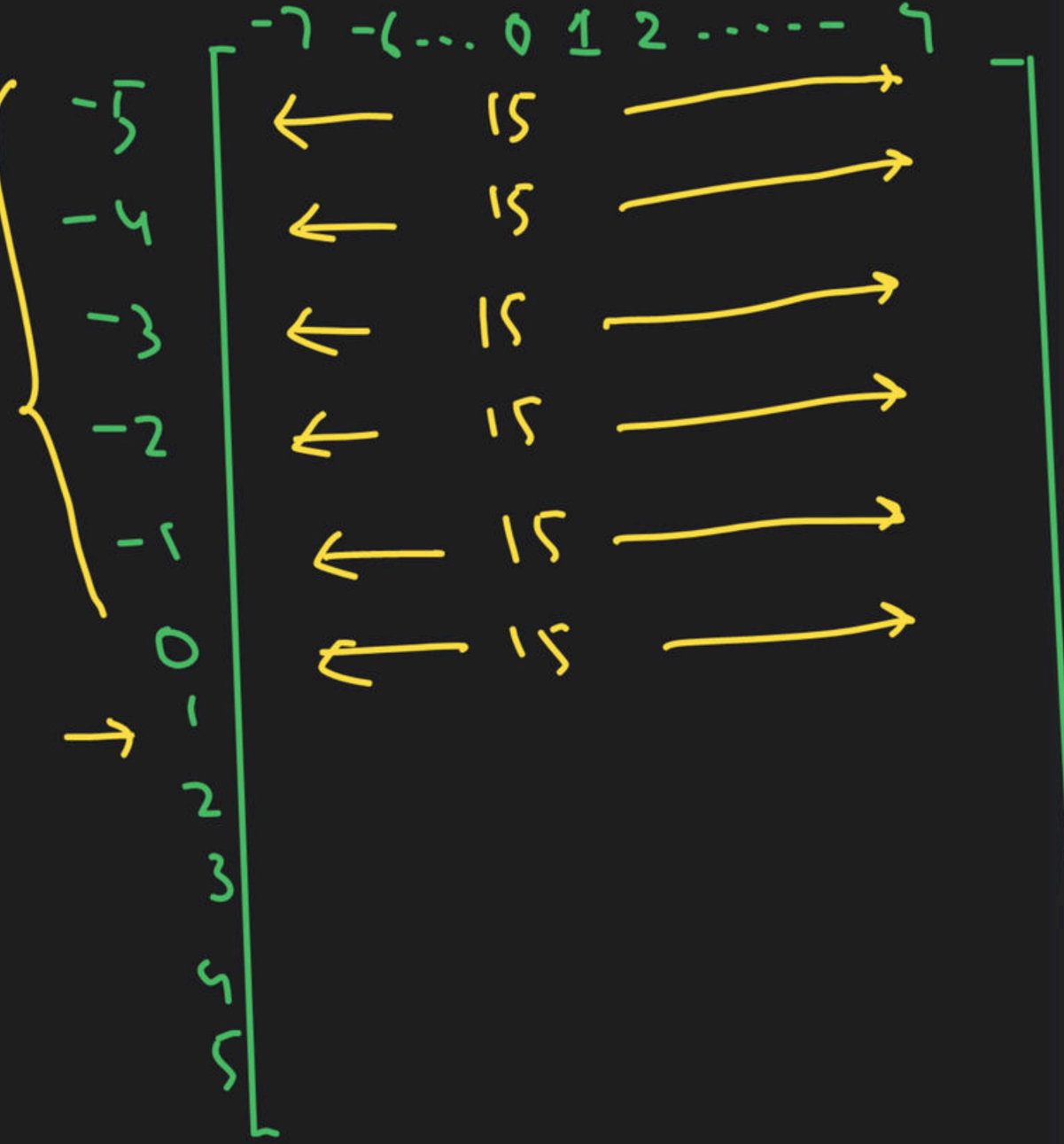
$A[-5..5][-7..7]$ $\quad 7-(-7)+1 = 15$

$w = 4$ bytes, $BA = 1000$

add(A[1][3])

a) Every index in this dimension represent = 15 ele.

now with index -5



2. RMO

$A[-5..5][-7..7]$ $\quad 7-(-7)+1 = 15$

$w = 4$ bytes, $BA = 1000$

add(A[1][3])

Rows already filled

index -5 to 0

$0-(-5)+1$

$= 6$ rows

unacademy

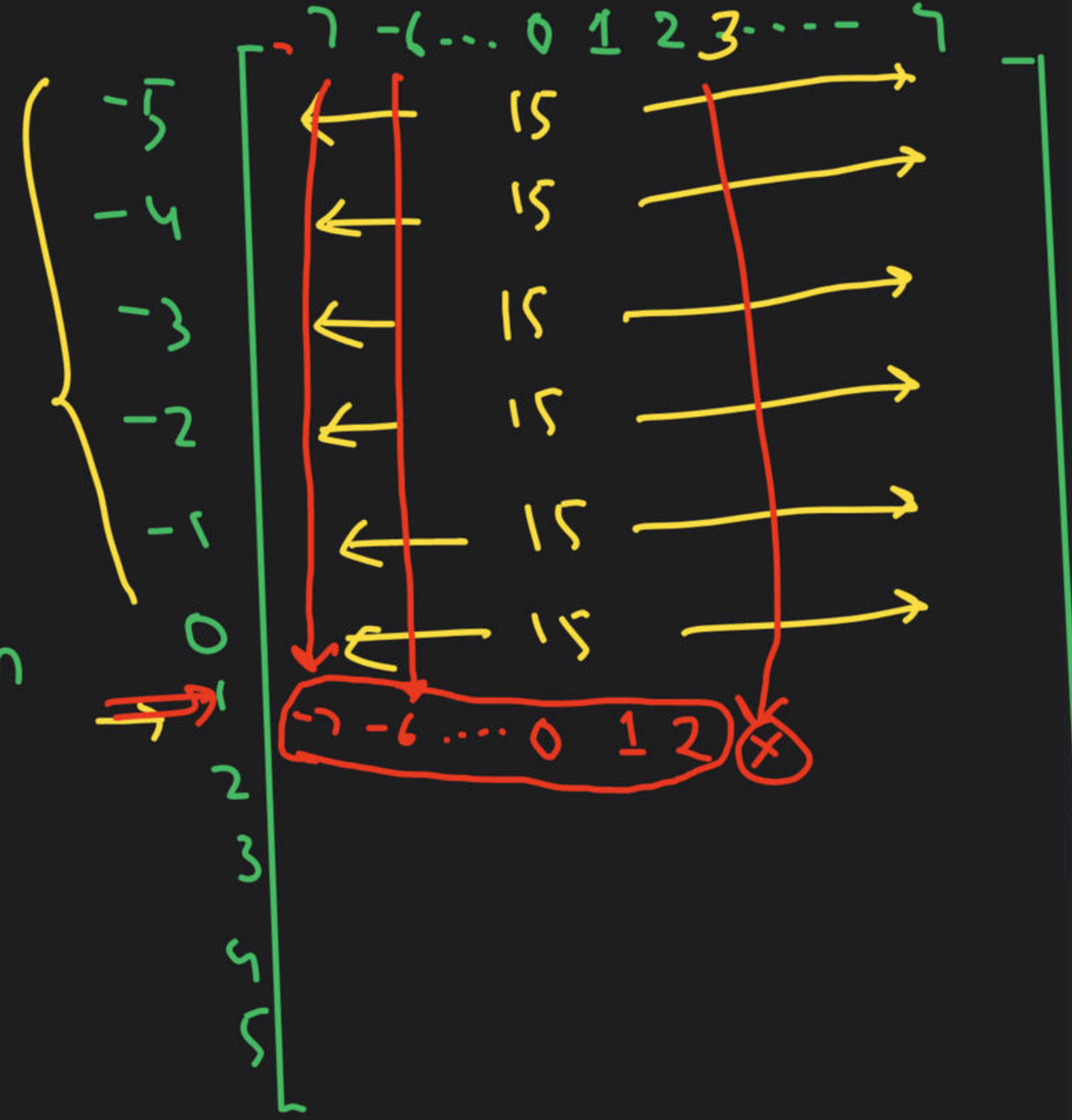
within row with index 1 — A_{13}

ele already filled before A_{13}

= Col index −7 to 2

= 2 − (−7) + 1

= 10 elements



After 6 rows & 10 elements, A_{13} is stored

Total ele. already filled before $A_{13} = 6 \times 15 + 10$

$= 90 + 10 = 100$ ele.

Memory already filled before $A_{13} = 100 \times 4 = 400$ Bytes

$\text{Add}(A_{13}) = 1000 + 400$

$= 1400$

A[M][N]

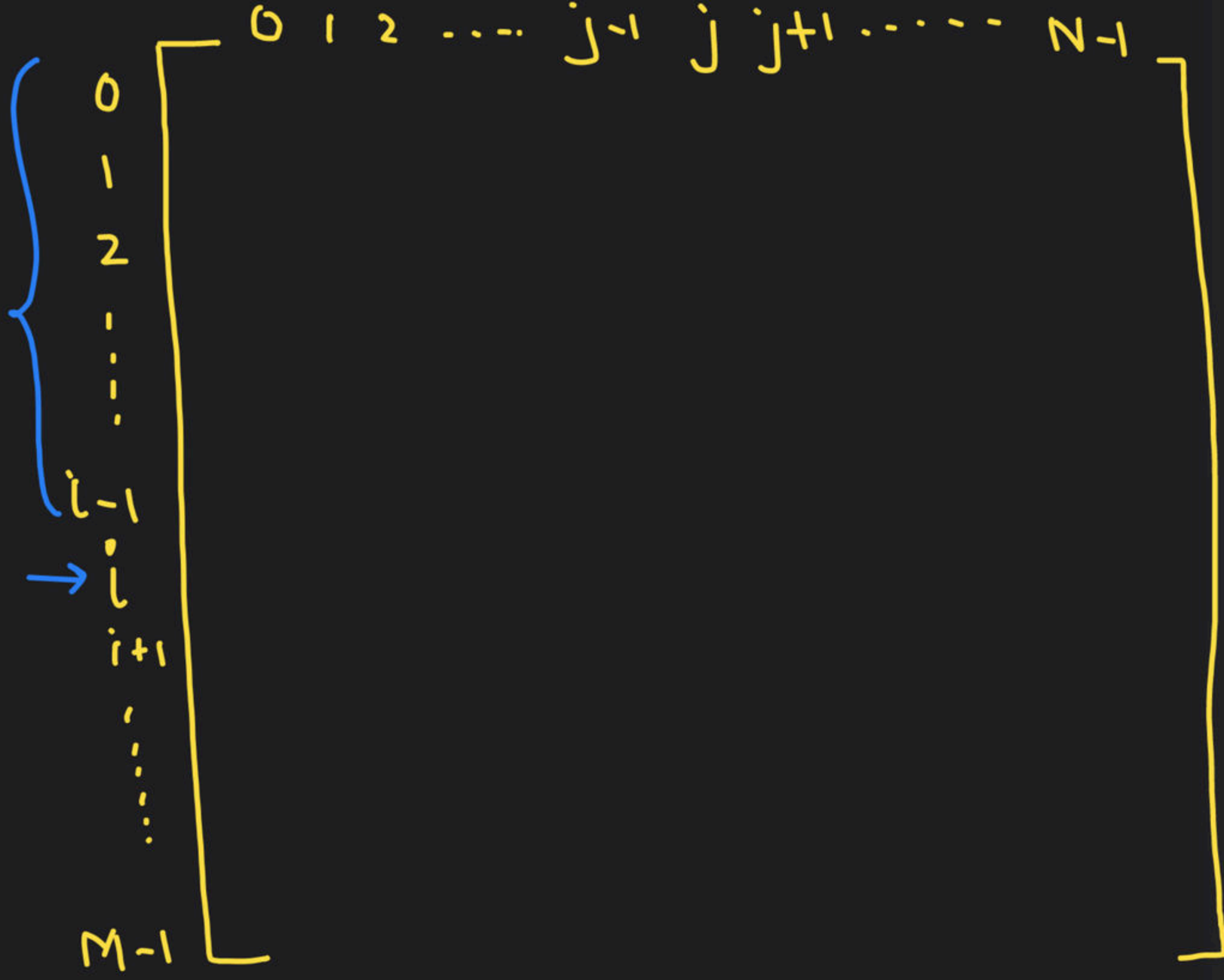
add(Aij)

rows already filled before row with index i

= index 0 to i-1

= i-1-0+1

= i rows



unacademy

A[M][N]

addr(Aij)

→ 0 to j-1

rows already filled before row with index i

= 'index 0 to i-1

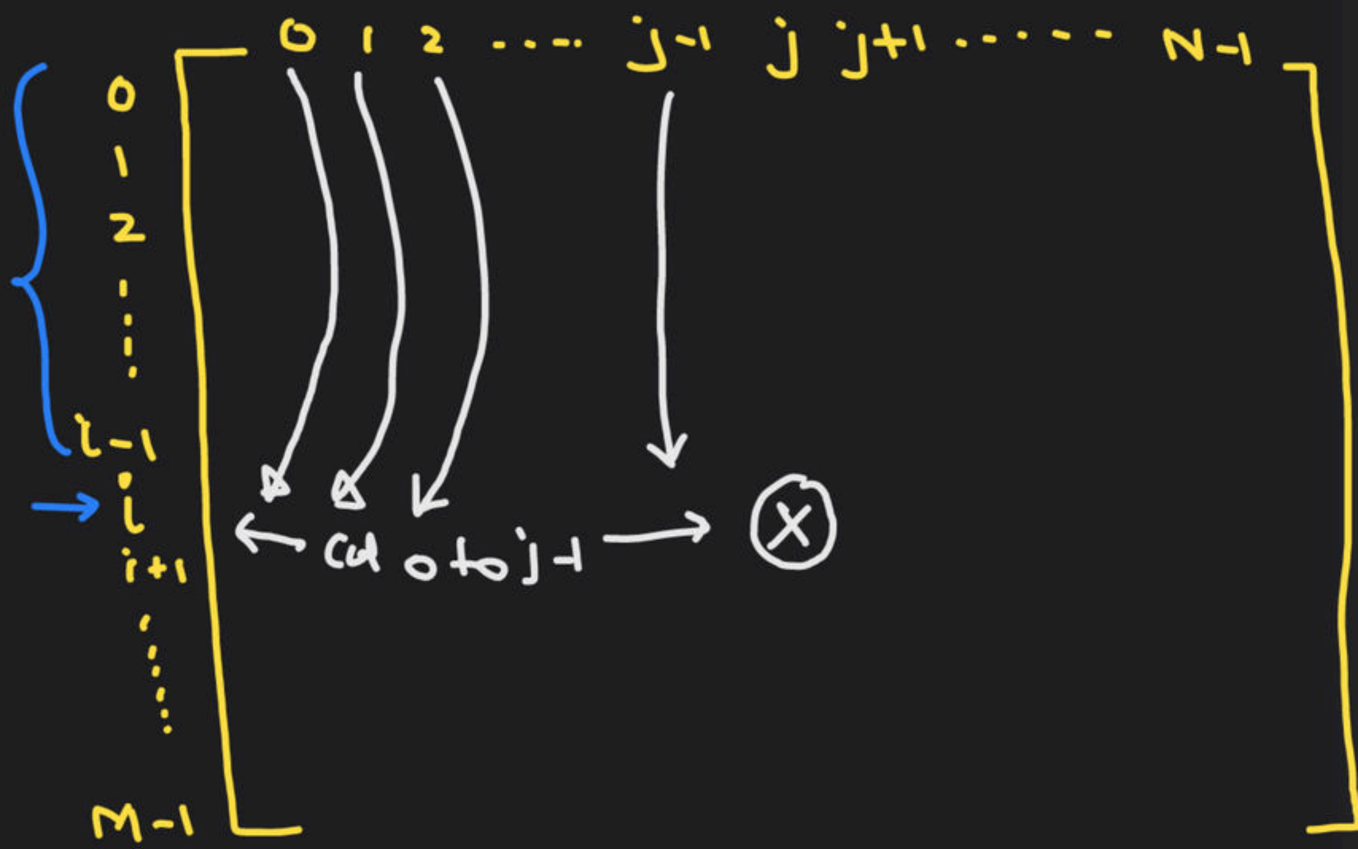
= i-1-0+1

= i rows

within row with index i, elements already filled before Aij

= col index 0 to j-1

= j-1-0+1 = j elements

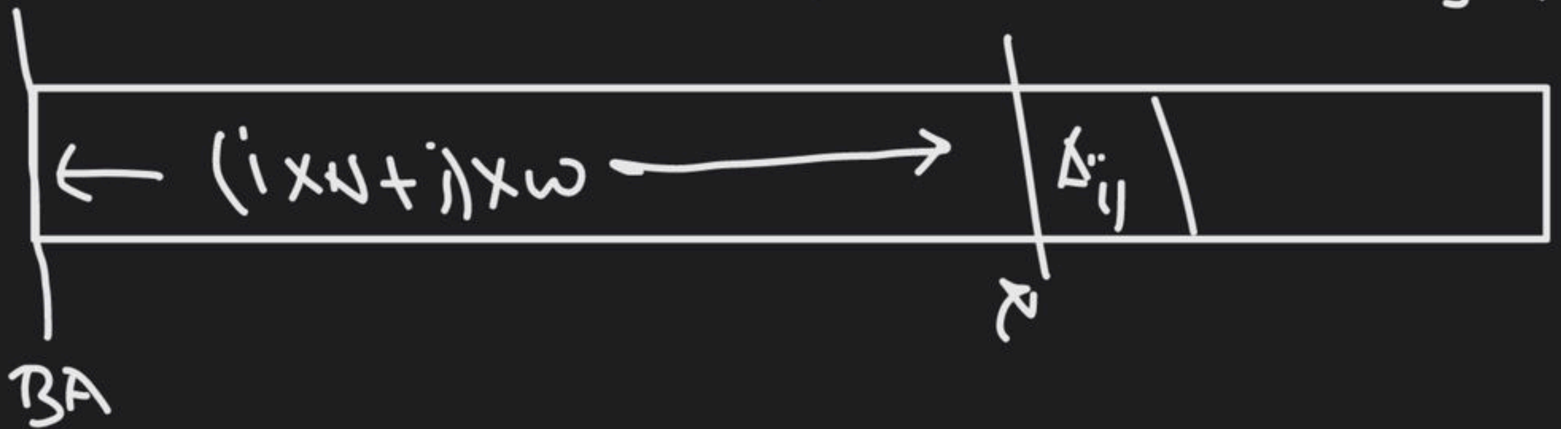


After i rows & j elements A_{ij} is stored.

Total ele. already filled before A_{ij}

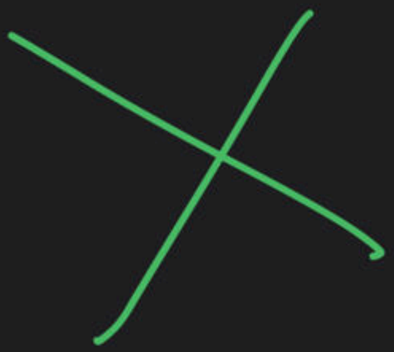
$$= (i \times N + j)$$

Memory already filled before A_{ij} = $(i \times N + j) \times w$ Bytes



$$add(A_{ij}) = BA + (i \times N + j)\omega$$

Ratna Nahi hai



Q/ $A[-20..20][-15..15]$ $15-(-15)+1 = 31$ $w = 2$ Bytes $BA = 1000$

Add $(A[2][1])$

Total ele. already filled before $A_{2,1}$

= $22 \times 31 + 21$

= 703 ele.

rows already filled

rows already filled = -20 to 1

= $1-(-20)+1$

= 22 rows

= 22×31

-15 to 5

$5-(-15)+1$

$\Rightarrow$ 21 elem

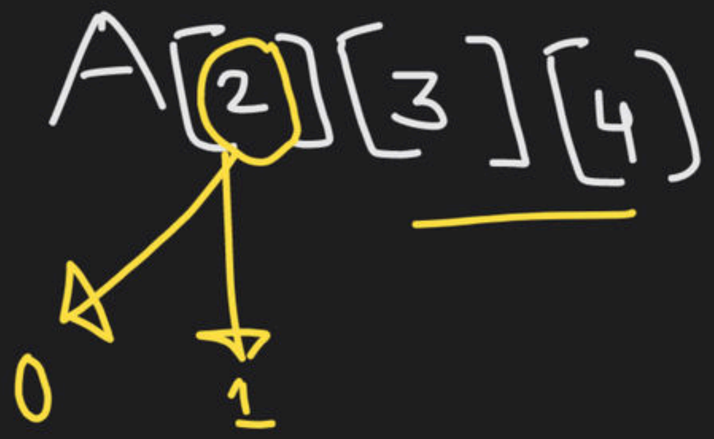
Memory already filled before A_{26} = 703 x 2 = 1406 Bytes

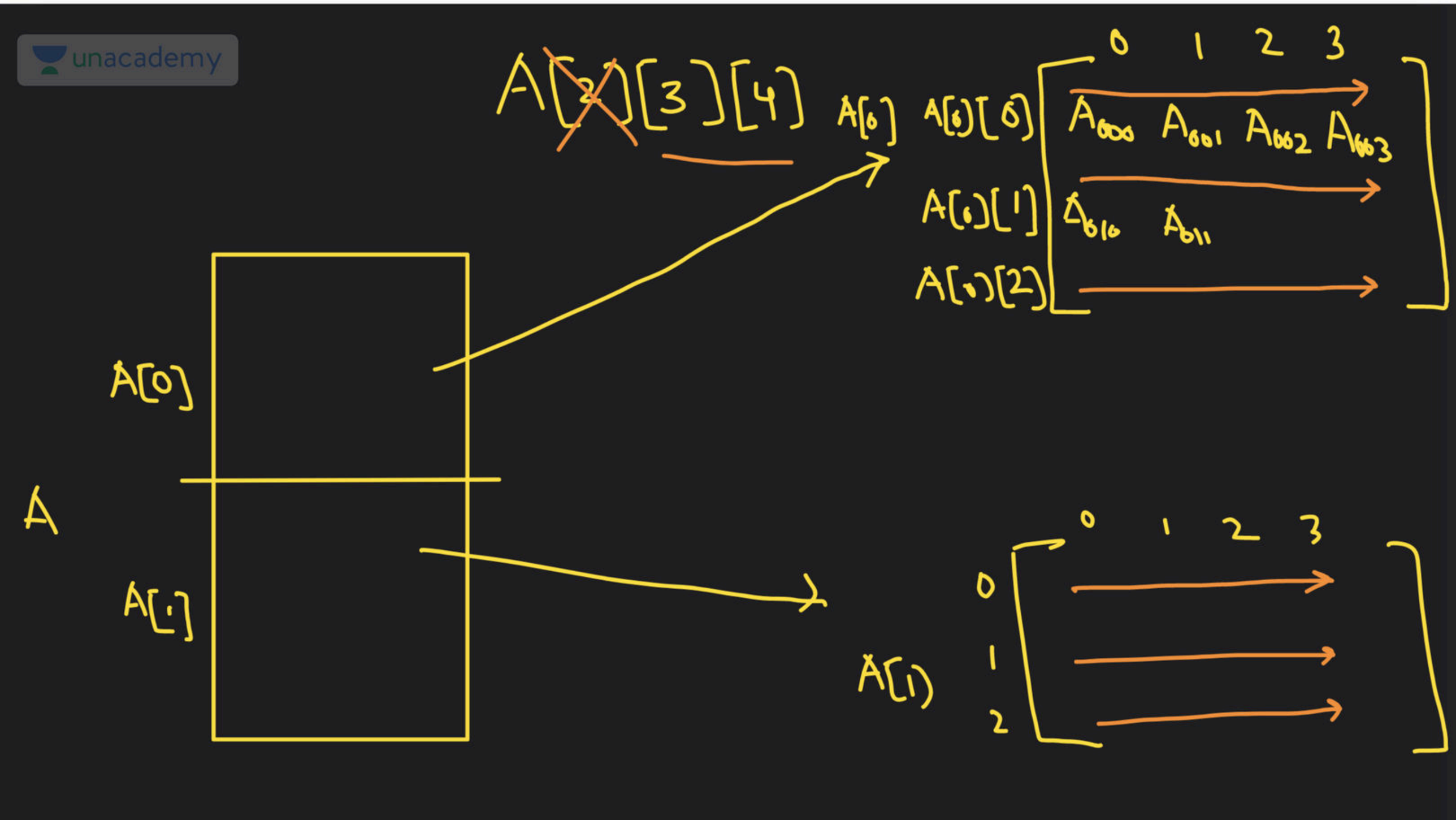
$$Add(A_{26}) = 1000 + 1406 = 2406.$$

unacademy

A[0]

A[1]




A[2][3][4]
A[6]
A[6][0]
A[6][1]
A[6][2]
0 1 2 3
A600 A601 A602 A603
A610 A611
A[0]
A
A[1]
A[1]
0
1
2
0 1 2 3

unacademy

A[2][3][4]

0

A_0

000 001 002 003

010 011 012 013

020 021 022 023

A_1

A_0

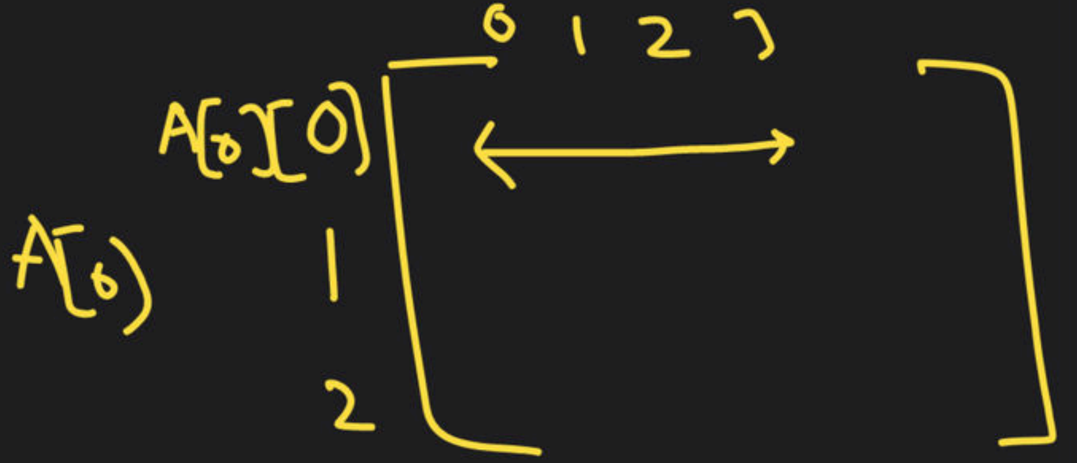
unacademy

A[2][3][4]

Every index in this dim = 3 x 4

Every index in this dim = 4

A[0][0] A[0] 0 1 2



A[1]

unacademy

RMO

BA = 1000

w = 2 Bytes

A[2][3][4]

add(A[1][2][3])

Total no. of ele. already filled

= 1x3x4 + 2x4 + 3

= 23 ele.

index already filled

0 to 0

⇒ 0-0th

= 1

⇒ 1 x3x4

index already filled

= 0 to 1

= 1-0th

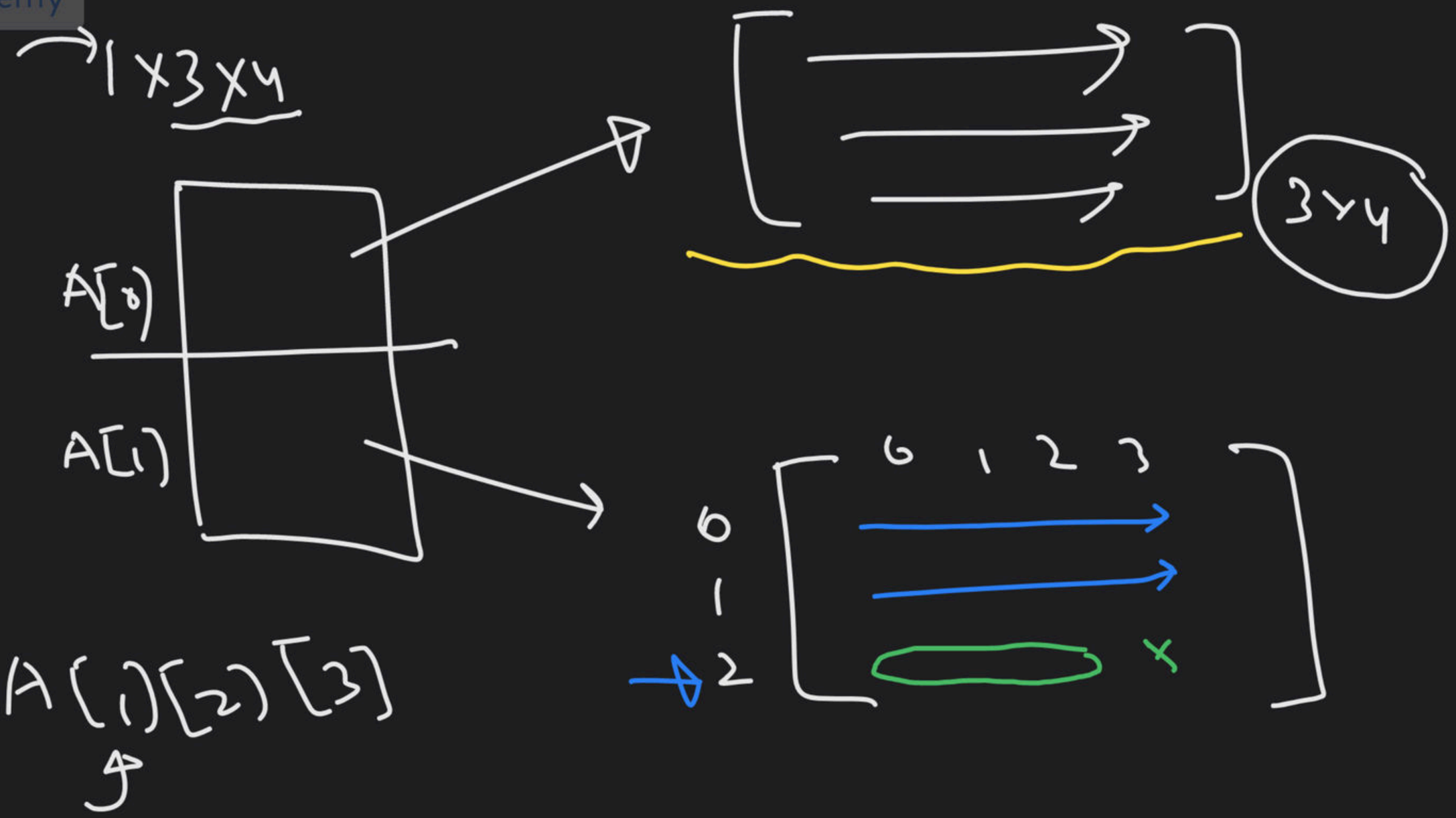
= 2

= 2x4

index already filled

= 0 to 2

= 2-0th = 3 ele



unacademy

THANK YOU!

Here's to a cracking journey ahead!