Agenda

lot of functions
↓

- * ✓ Fit bit case study => 2D-array
- * ✓ 2D-Axis => axis=0, axis=1 { Assignments
- * ✓ Sorting ✓ => ⟹ sort ↓
- * ✓ Matrix Multiplication => ✓ [ ] => dot, mat, @
- * ✓ Vectorization ⟹
- * ✓ Broadcasting =>

Recap

=>

1. 2D array → [[ ] , [ ]]

✓ Transpose →

✓ Indexing → A[0, 1]

✓ slicing → A[: , :]

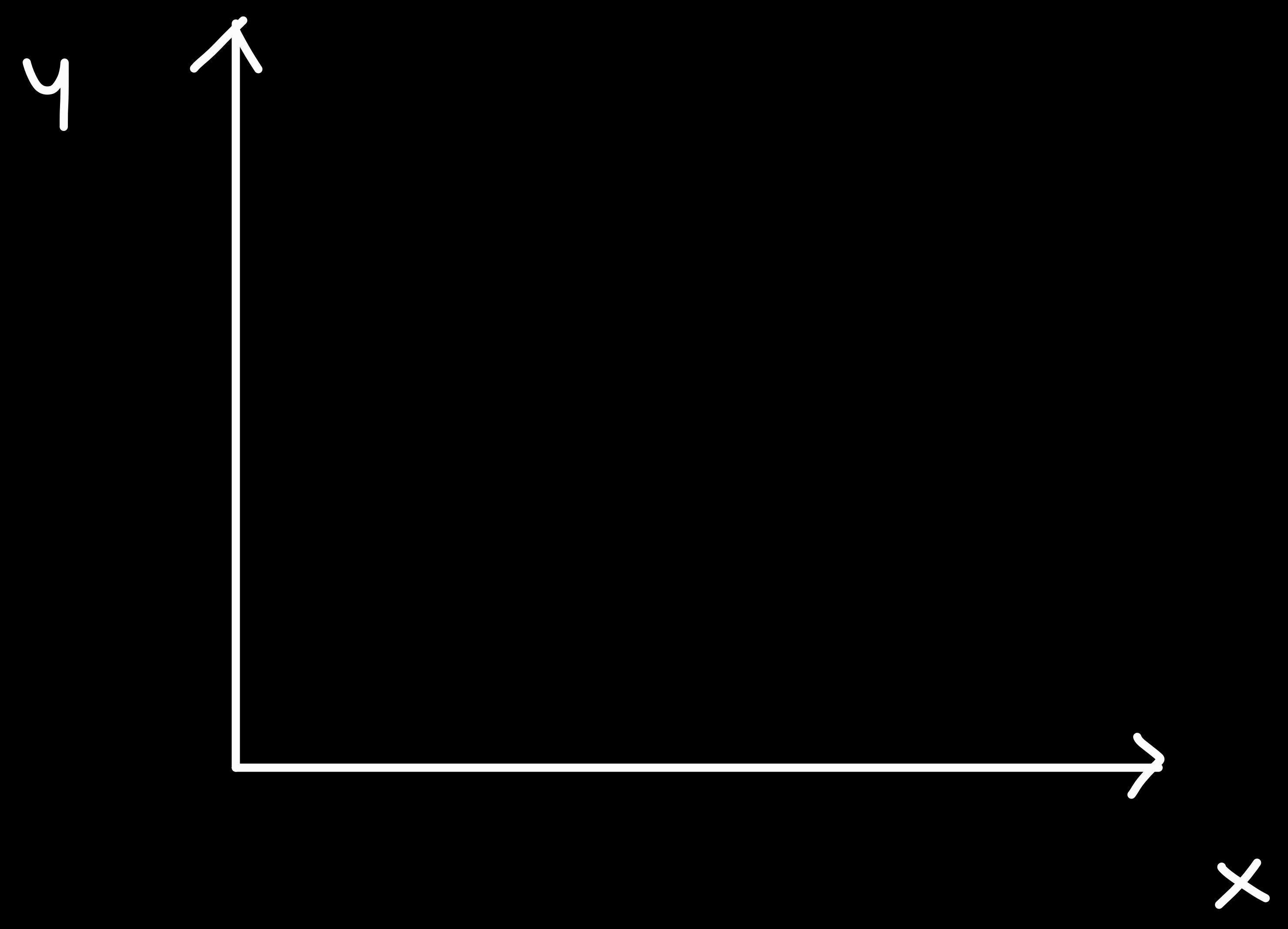
m w

✓ Fancy Indexing [masking] => 1D array

2. Logical operations → any(), all(), where()

Aggregate functions → sum, min max, mean

⇒



①

```
a = np.array([1, 2, 3, 4])
```

```
a.shape ↵
```

$$\boxed{(4,\ )}$$

$$\left[\begin{array}{l} \overset{\swarrow (4,)}{\checkmark\,[\ 1\ \ 2\ \ 3\ \ 4\ ]} \\ \overset{(4,)}{\checkmark\,[\ 5\ \ 6\ \ 7\ \ 8\ ]} \\ \overset{(4,)}{\checkmark\,[\ 9\ \ 10\ \ 11\ \ 12\ ]} \end{array}\right]$$

②

```
a = np.arange(1, 13).reshape(3,4)
```

```
a.shape ↵
```

$$(\underset{\uparrow}{\boxed{3}}\ 4)$$

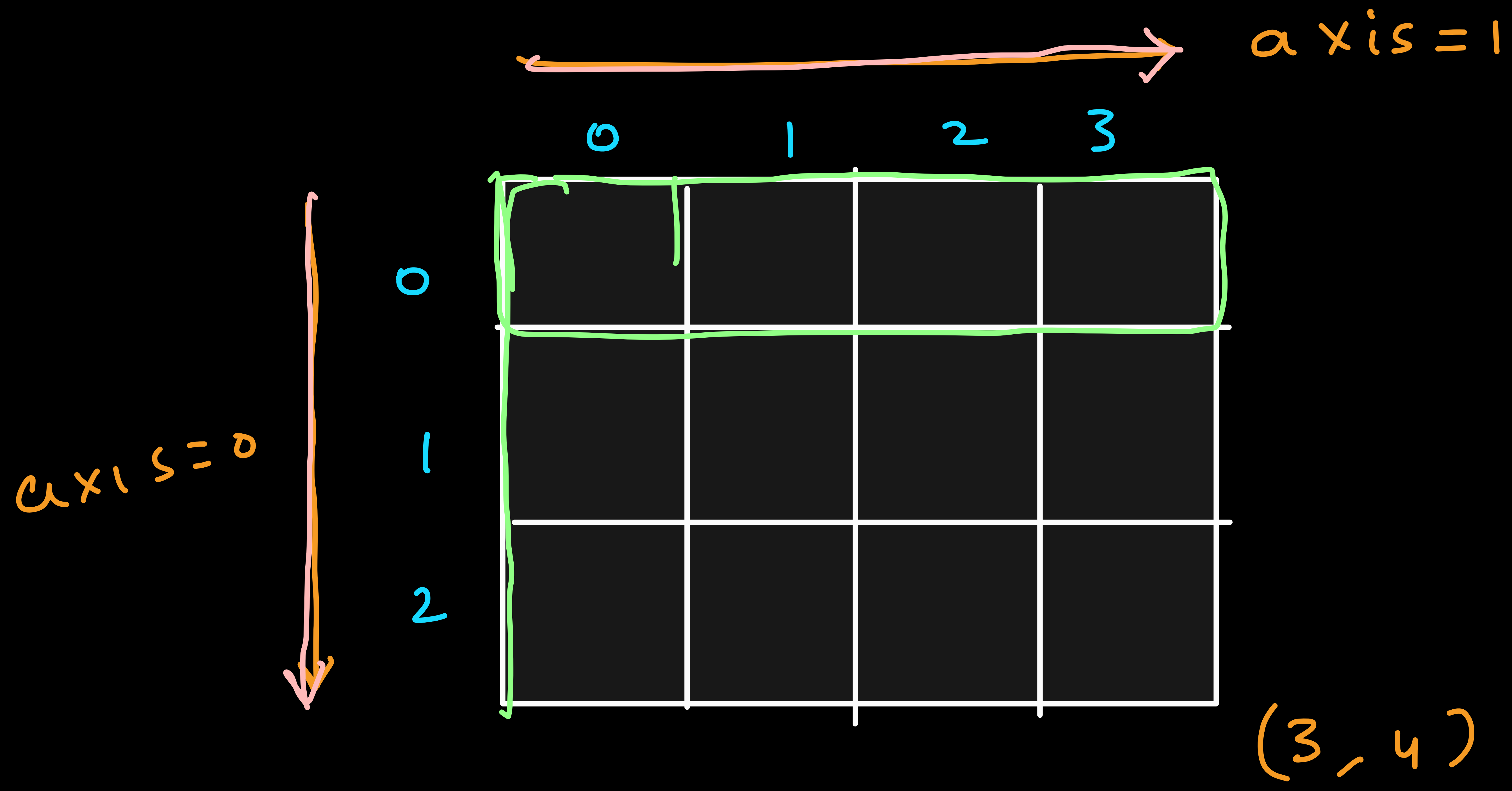
$$\left[\begin{bmatrix}1 & 2\\ 3 & 4\end{bmatrix}, \begin{bmatrix}5 & 6\\ 7 & 8\end{bmatrix}\right] \Rightarrow (\textcircled{2}, 2, 2)$$

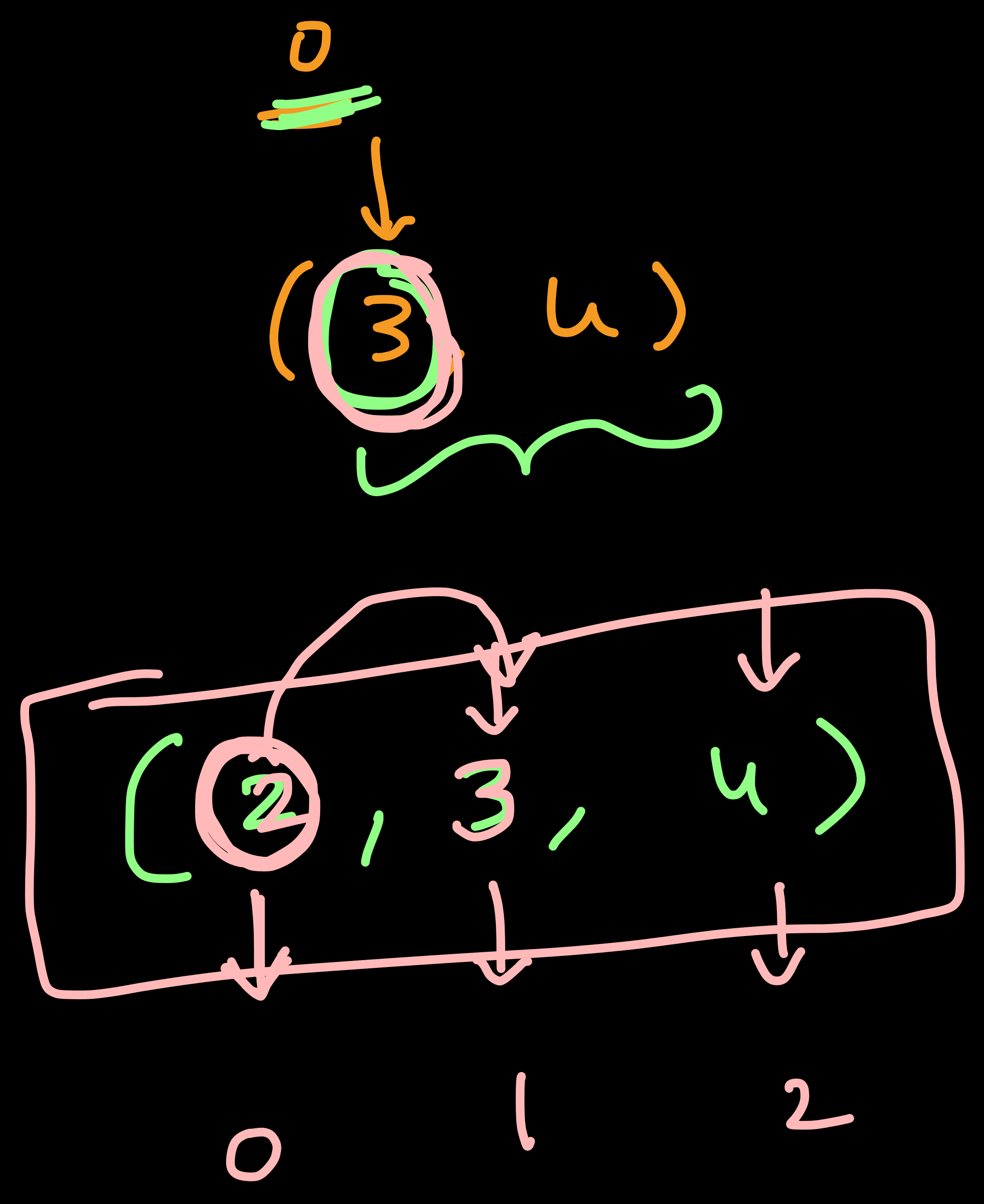
$$\downarrow$$

$$(2, 2)$$

$$\begin{array}{ccc} 0 & 1 & 2 \\ (\textcircled{2}, & 3, & 4) \\ \downarrow & \downarrow & \downarrow \\ \text{axis}=0 & \text{axis}=1 & \text{axis}=2 \end{array}$$

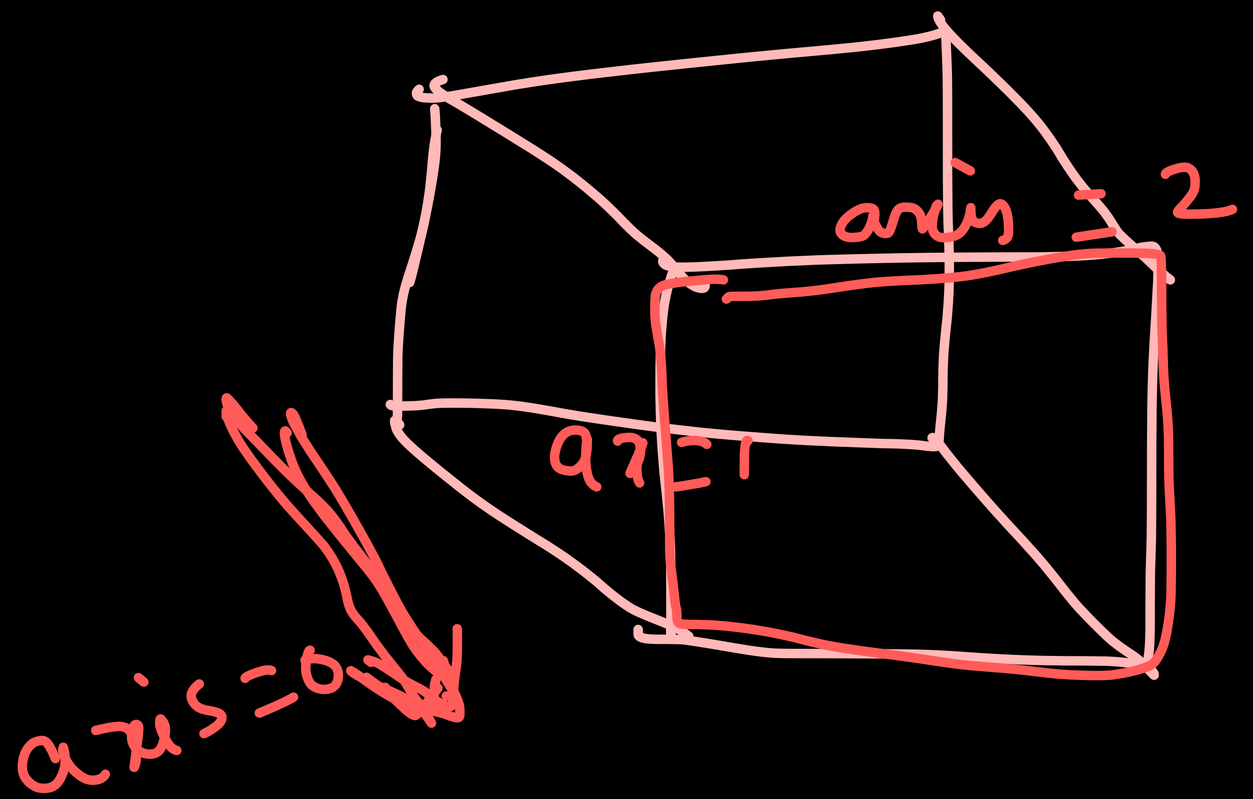
$$\left[\overset{\downarrow\ 3,4}{\begin{bmatrix}1, & 2, & 3, & 4\\ 5, & 6, & 7, & 8\\ 9, & 10, & 11, & 12\end{bmatrix}}, \overset{\downarrow\ 3,4}{\begin{bmatrix}1, & 2 & 3 & 4\\ 5 & 6 & 7 & 8\\ 9 & 10 & 11 & 12\end{bmatrix}}\right]$$





running vertically downwards accross rows (axis=0) =>

runing horizontally across columns (axis=1)

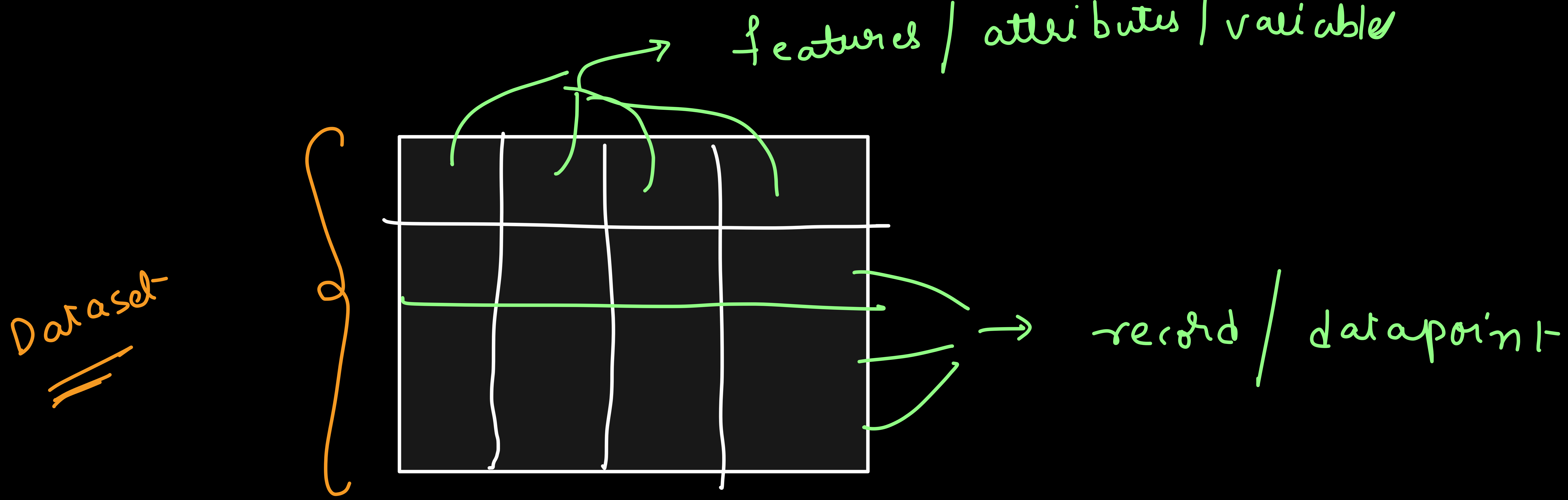

axis = 2
ax=1
axis=0

✓ Use case: Fitness Data Analysis

Data Analysis

Imagine you are a Data scientist at Fitbit

You’ve been given a user data to analyse and find some insights which can be shown on the smart watch



⇒ What kind of questions can we answer using this data?

1. ✓ How many records & features are there in the dataset → shape

2. Avg stepcount

3. which day the stepcount was highest/lowest?

4. How daily activity affects sleep and mood?

5.

	0	1	2	3	4	5
Step count :	[1000,	2000,	1500,	7500,	800,	7500]
Date :	[1-1-2025,	2-2-2025,	15-1-2025,	16-2-2025	. . . .	]

Date [np argmax(step-count)] ← ③

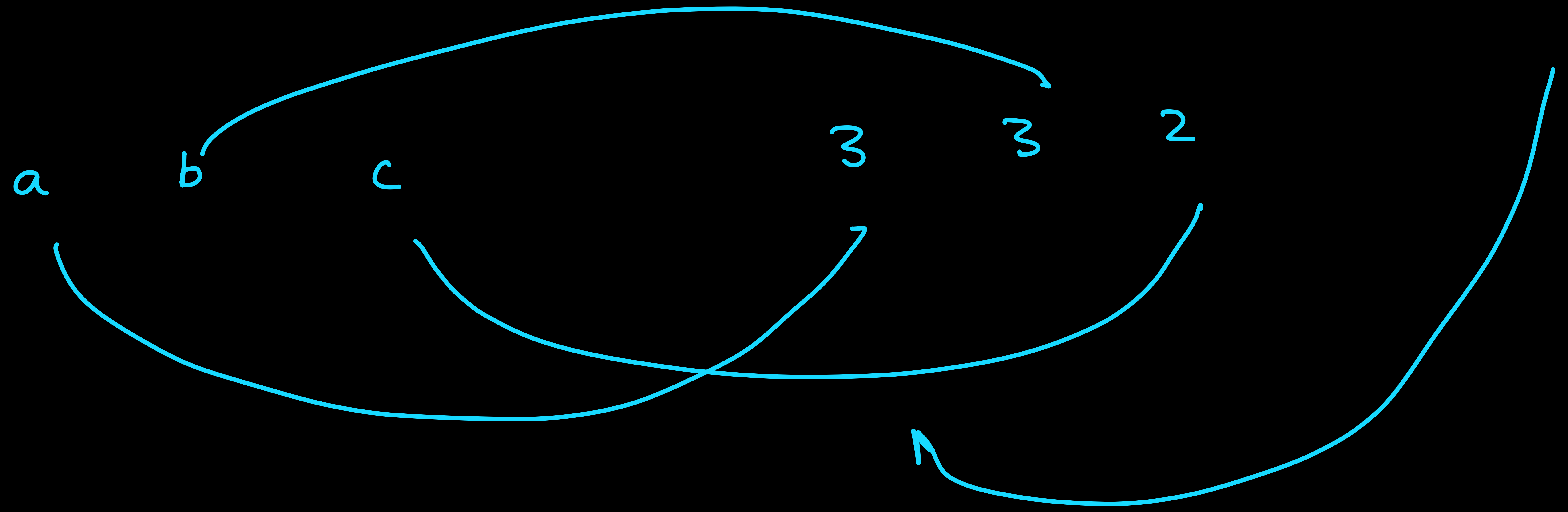
16-2-2025 ✓✓

arr = ['a', 'b', 'a', 'b', 'c', 'a', 'b', 'c']

np.unique(arr,

return_count =

True)



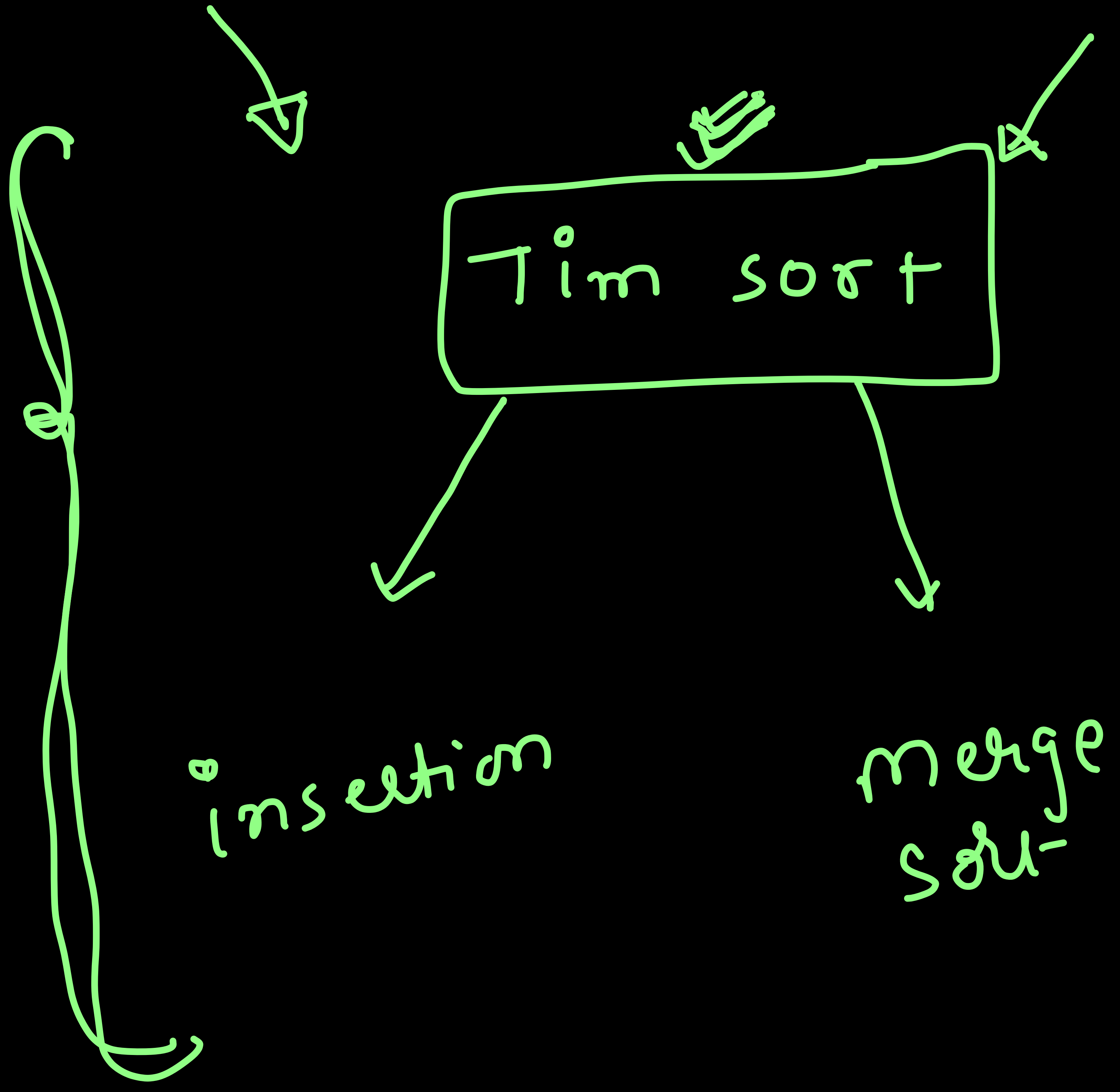
Sorting

Bubble, insertion, merge - - - - - - - - -

O(n log n)

sort

Time & space


Tim sort

insertion

merge sort

Matrix multiplication

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} * B = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} = \begin{bmatrix} 1 & 4 & 9 \\ 16 & 25 & 36 \\ 49 & 64 & 8 \end{bmatrix}$$

3×3 3×3

$A \times B \Rightarrow 3 \times 3$

$$\begin{bmatrix} 1*1+2*4+3*7 & 1*2+2*5+3*8 & 1*3+2*6+3*9 \\ 4*1+5*4+6*7 & - \quad - \; - \; - \; - & \\ & & \end{bmatrix}$$

Math for ML

1 D Array → $\boxed{\text{Vector}}$ ✓ →

2 D array → Matrix

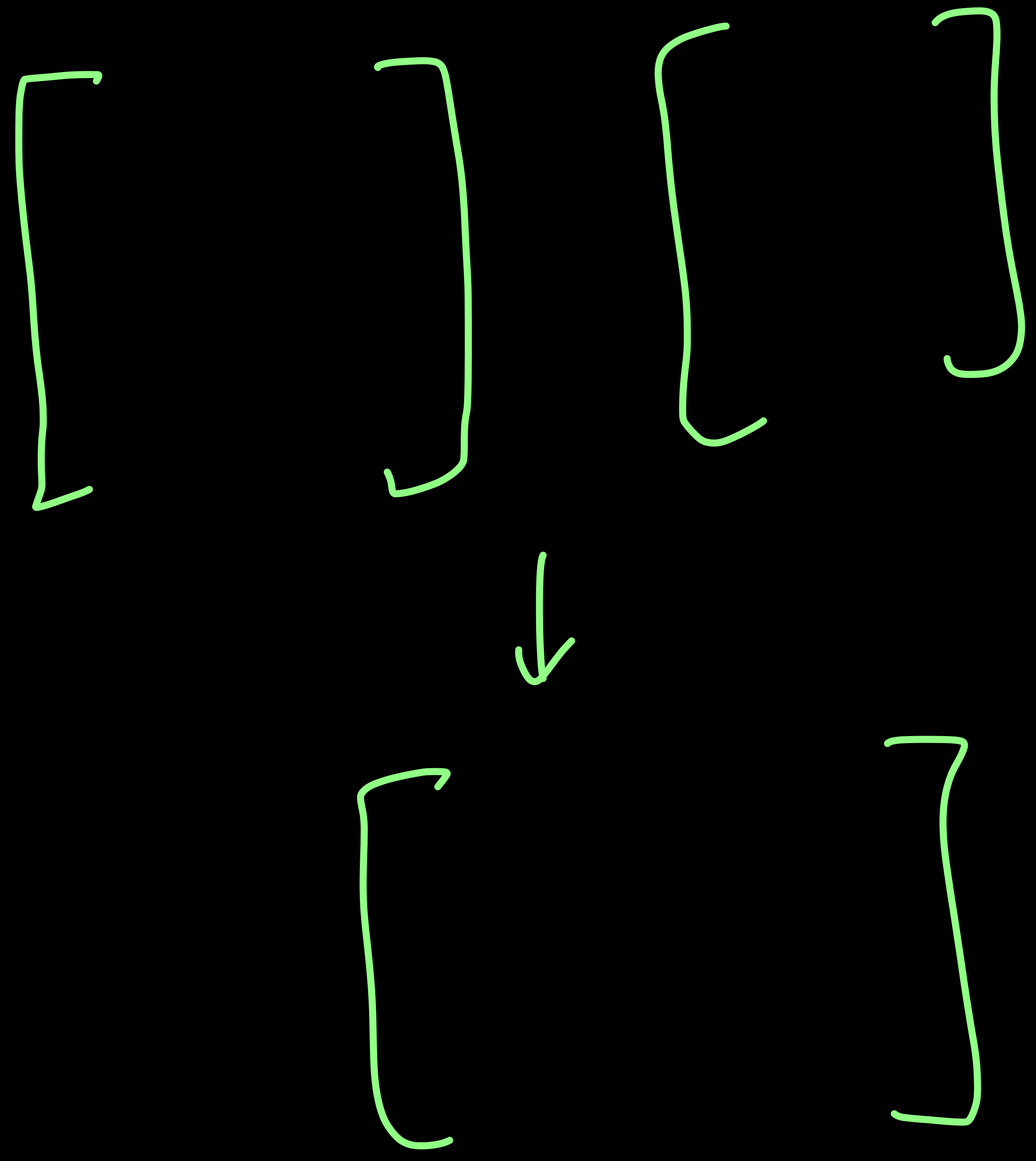
3 D array → Tensor

$$[1\ \ 2\ \ 3\ \ 4] * [1\ \ 2\ \ 3\ \ 4]$$

↓

Scaler

$\boxed{1+4+9+16}$ ✓✓



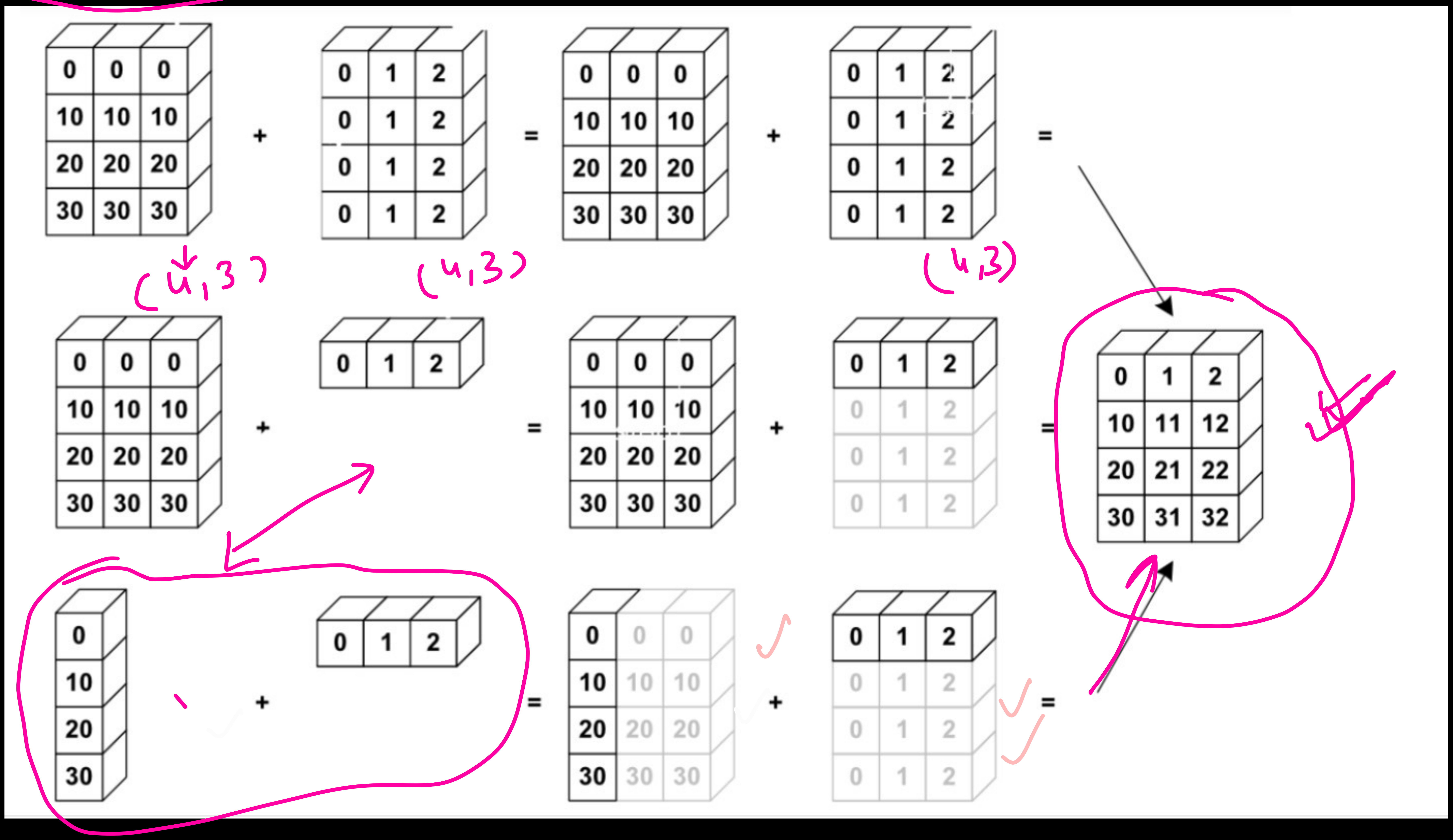
Broadcasting

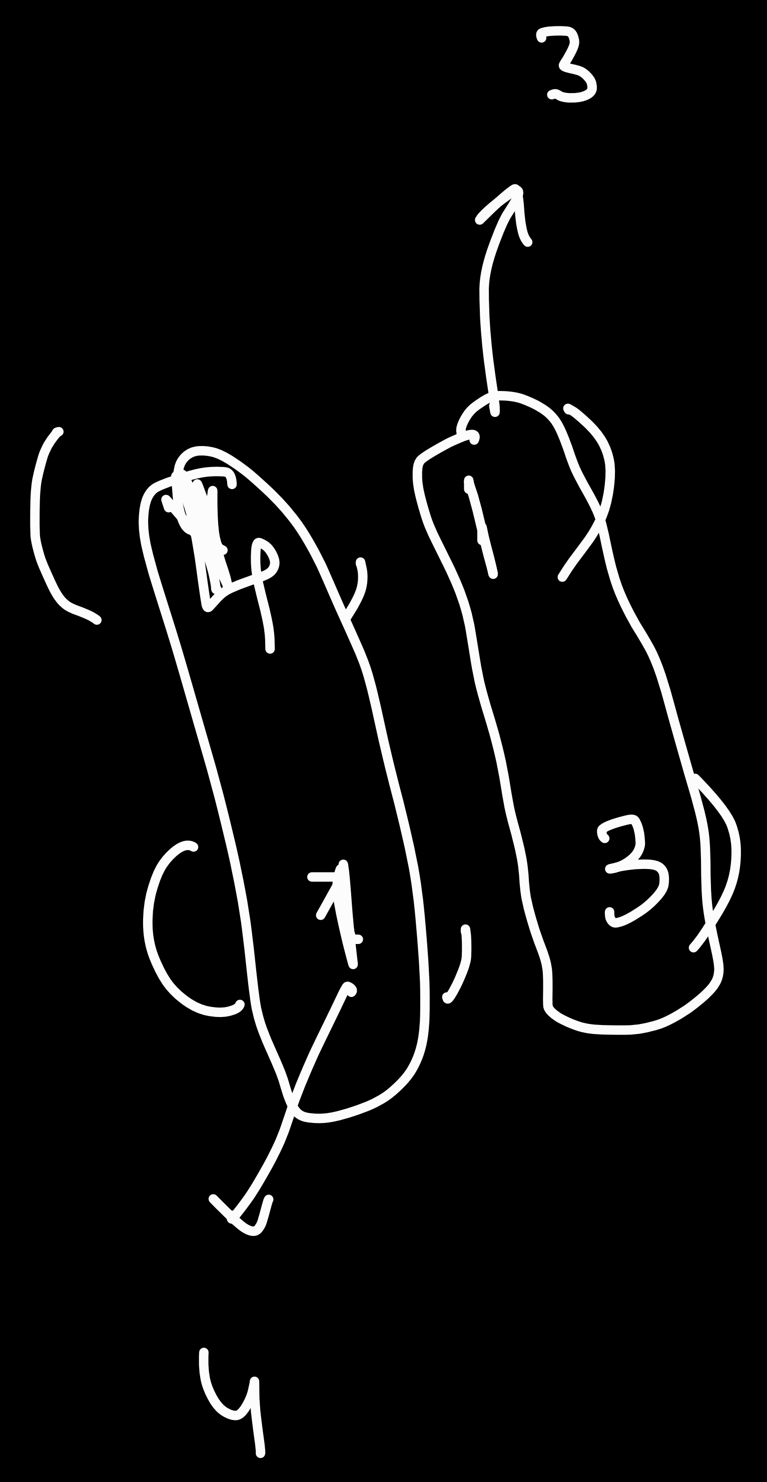
↳ Rules

1

2

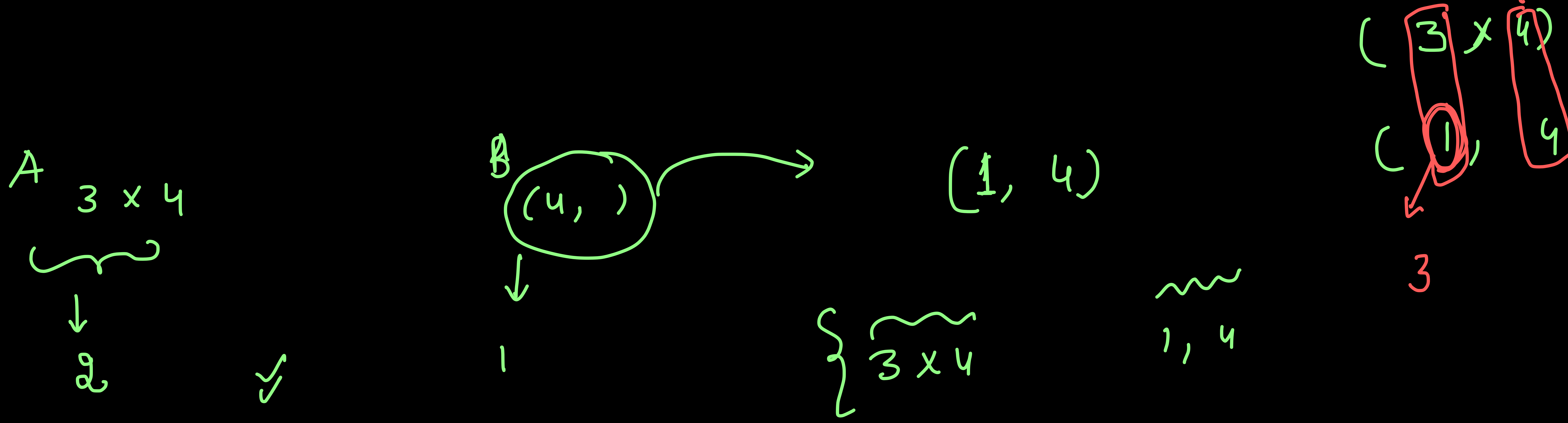
3





Rule 1 : If two array differ in the number of dimensions, the shape of one with fewer dimensions is padded with ones on its leading(Left Side).

Rule 2 : If the shape of two arrays doesnt match in any dimensions, the array with shape equal to 1 is stretched to match the other shape.



$$\begin{bmatrix} 0 & 0 & 0 \\ 10 & 10 & 10 \\ 20 & 20 & 20 \\ 30 & 30 & 30 \end{bmatrix}$$

$(4,3)$

$\downarrow$

$$\begin{bmatrix} \begin{bmatrix} 0 & 1 & 2 \end{bmatrix} \end{bmatrix}$$

$\downarrow$

$\downarrow \quad (\cancel{1},3)$

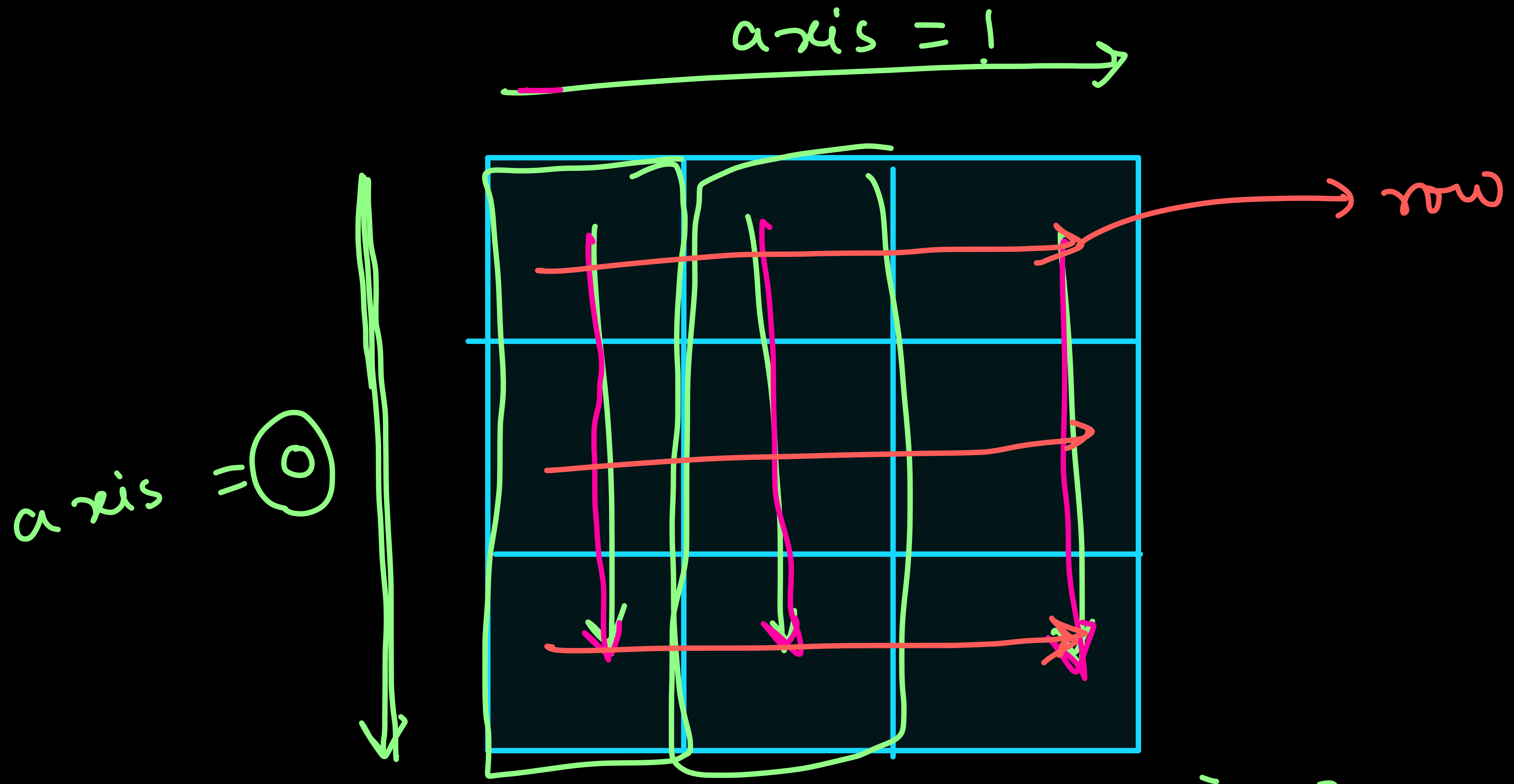
$$\begin{matrix} (\boxed{4}, & \boxed{3}) \\ (\boxed{\cancel{1}}, & \boxed{3}) \\ 4 & \end{matrix}$$

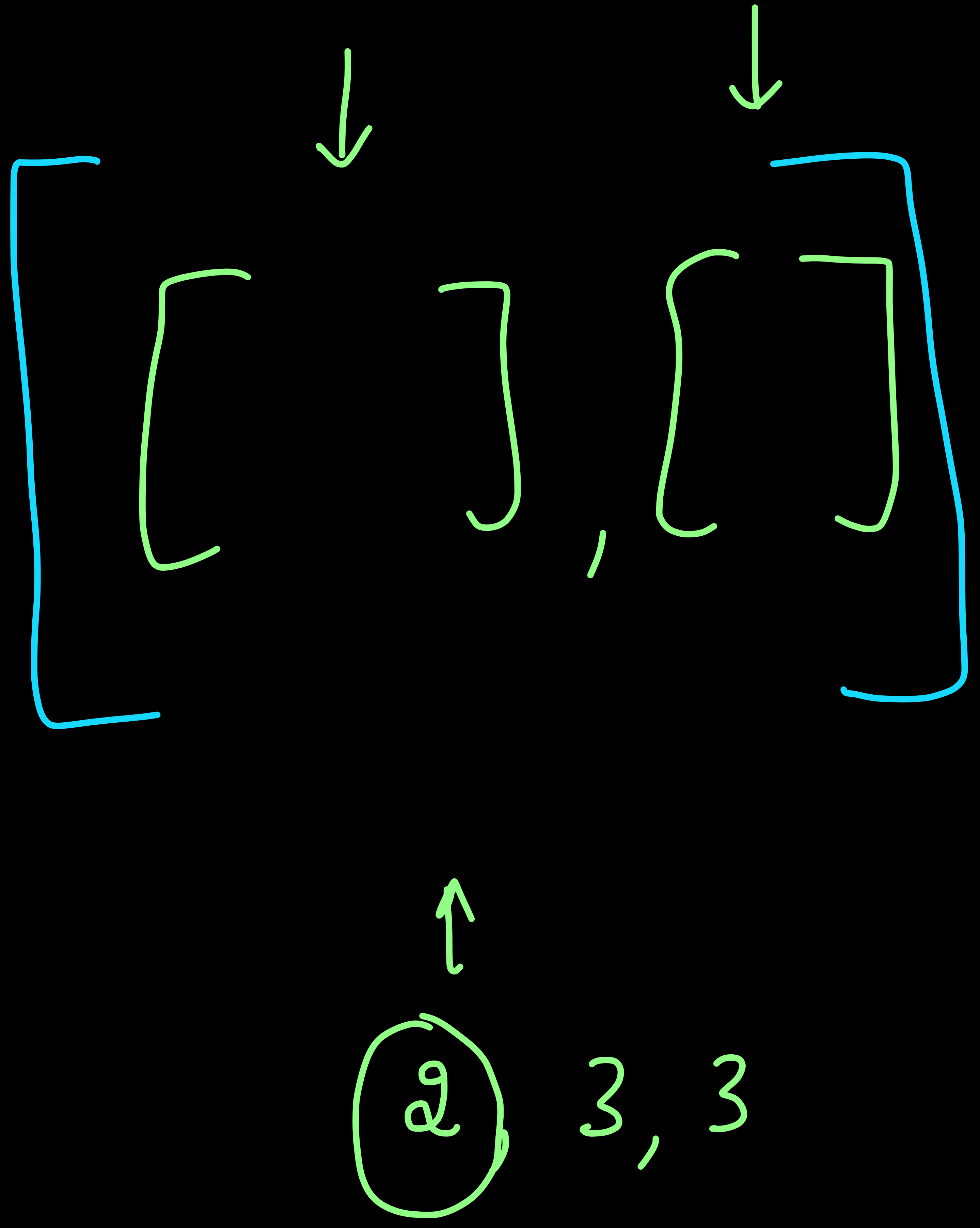
$$\textcircled{1} \quad 2\begin{bmatrix} 0 & 0 & 0 \\ 10 & 10 & 10 \\ 20 & 20 & 20 \\ 30 & 30 & 30 \end{bmatrix} + \begin{array}{c} 1 \\ \begin{bmatrix} 0 & 1 & 2 \\ 0 & 1 & 2 \\ 0 & 1 & 2 \\ 0 & 1 & 2 \end{bmatrix} \end{array} \Rightarrow \checkmark$$

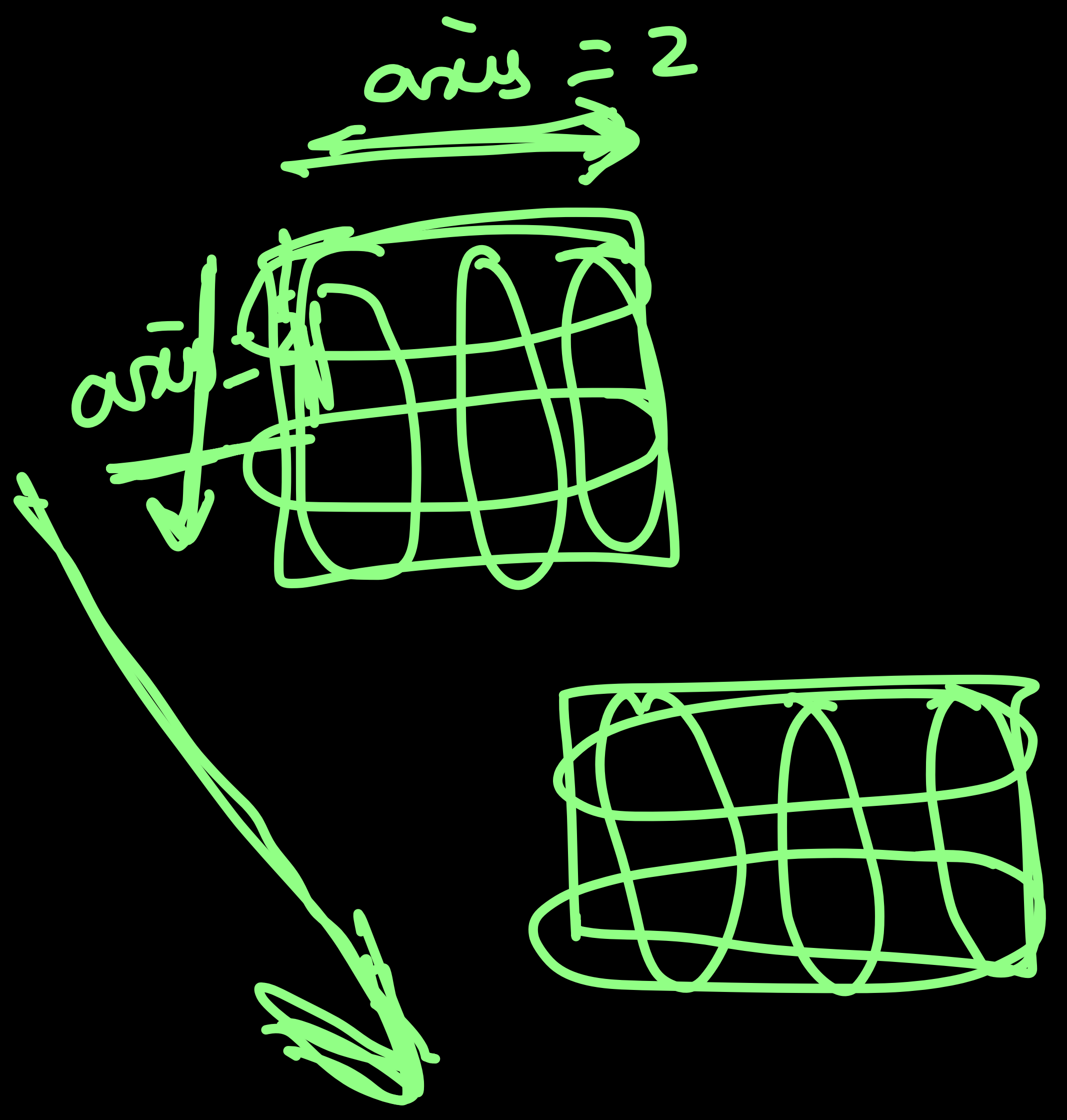
$A_{3 \times \boxed{4}}$ $B_{\boxed{3} \times 4}$ ⇒ $A \times B \Longrightarrow$?

$4 \; != 3$ ✗

✓ $$\begin{array}{cc} (\boxed{3}, & \boxed{4}) \\ & \downarrow \\ (3, & 1) \end{array} \rightarrow (3, 4)$$


axis = 1
row
axis = ⓪


② 3, 3


axis = 2
axi

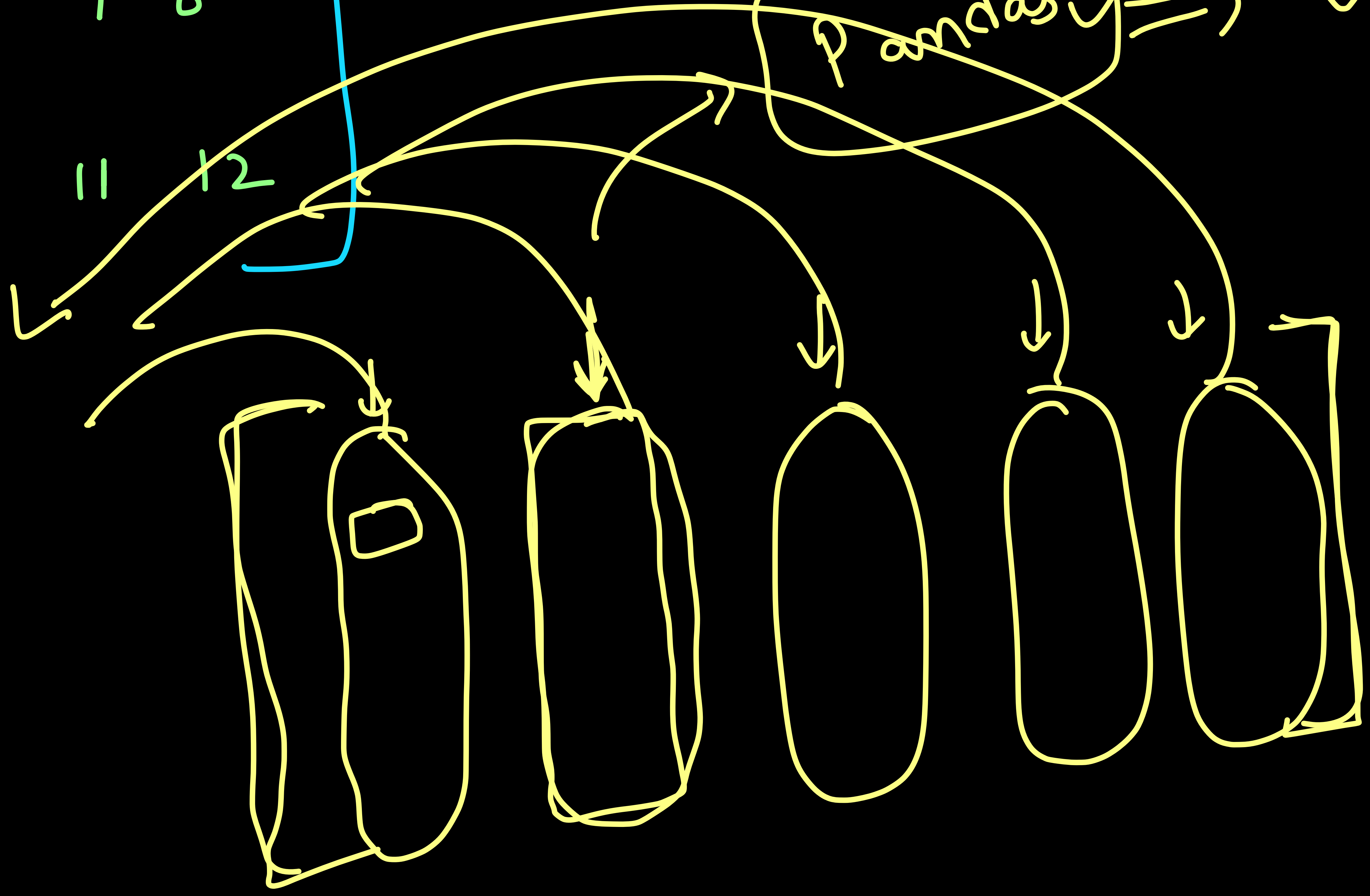
Numerical ✓ ⟹ numpy

$$\begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \end{bmatrix}$$

→ Geeks for geeks

→ Leetcode ✓ → { E, M, H ✓ }

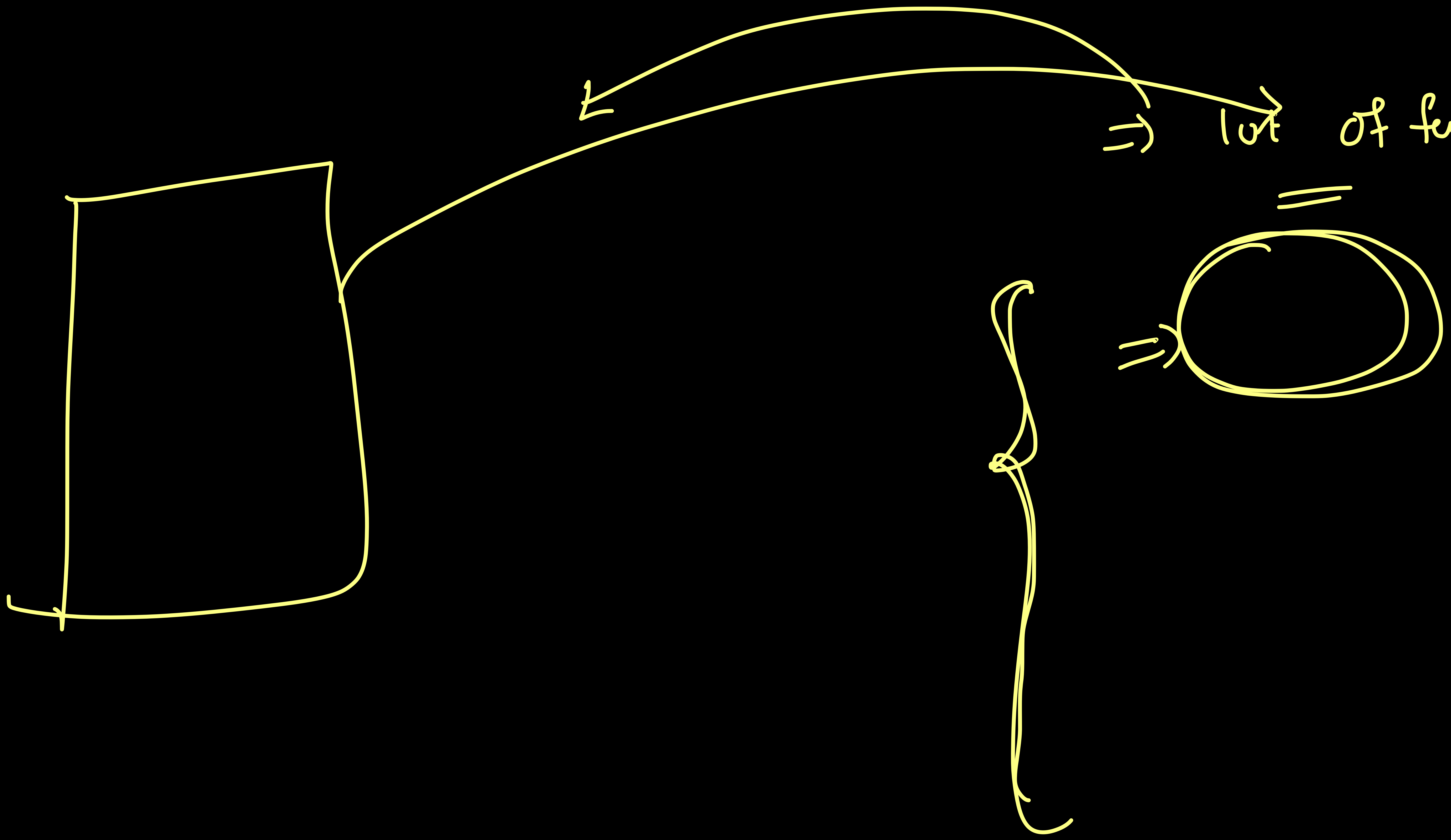
pandas ✓ ⟹ ✓



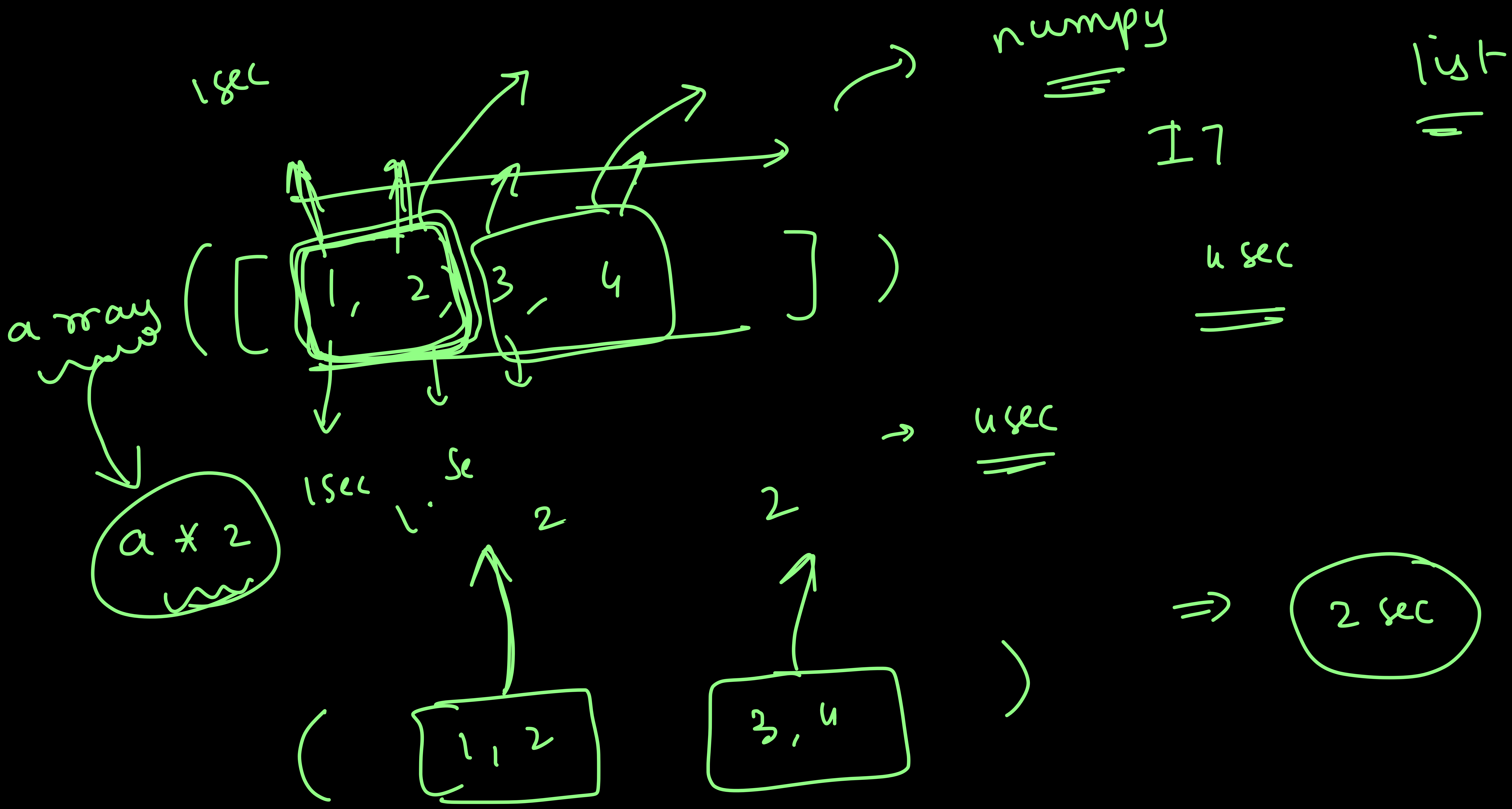
np.sort(axis = 0)

list

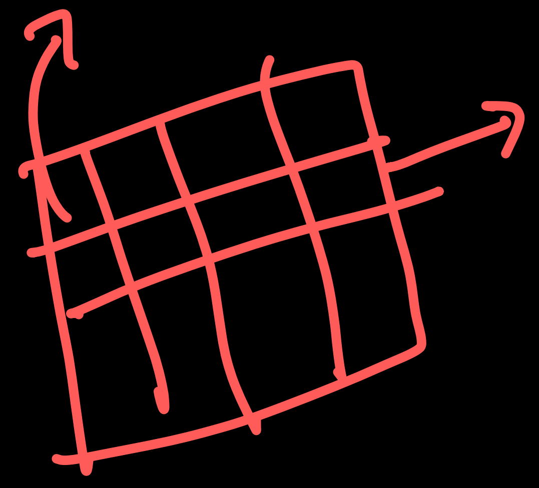
tuple



=> lot of function


1sec
numpy
list
I7
array
([1, 2, 3, 4])
4 sec
4sec
a * 2
1sec
1. Se
2
2
(1, 2 3, 4)
2 sec

Data Analysis ⇒ O O



✓ record | data

Sorting → axis = 0 → column

axis = 1 → rows

✓

✓

→ Matrix multi → dot ( ) ✓, matmul ✓, @ ✓

→ Vectorisation → element wise

higher order function

→ vectorize