Your data gets stored in the form of Numpy Array in Numpy library and not in the form of DataFrame

Numpy Array is partially similar to C++ or C Static array

int arr[5][5]; ⟶ C/C++ Static array ⟶

A Numpy array is somewhat similar to static C/C++ static arrays. but at the same time, a numpy array shares several features with Pandas DataFrame

Apart from this a Numpy array supports all the functions of Matrix Algebra (LINEAR ALGEBRA) → Matrix Multiplication, Matrix Addition, Transpose, Inverse of a matrix

$$5 \times 3 = 15$$

$$3 \times 5 = 15$$

$$\left.\begin{array}{l} 5 \times 3 \\ 3 \times 5 \end{array}\right\} = 15$$

$$\left.\begin{array}{l} 3 + 5 \\ 5 + 3 \end{array}\right\} = 8$$

$$3 - 5 = -2$$

$$5 - 3 = +2$$

$$\frac{3}{5} \qquad \frac{5}{3}$$

$$A \cdot B \neq B \cdot A$$

$$A + B = B + A$$

BODMAS

x, y

$x + y$

$x \cdot y$

$\frac{x}{y}$ $x - y$

MATRIX ALGEBRA

- $A + B$
- $A \cdot B$
- $A - B$

$\boxed{\frac{A}{B} \times}$ $\times$

$A^{-1} = \cancel{\frac{1}{A}}$

$5^{-1} = \frac{1}{5}$

$A.$

$$\begin{bmatrix} [1 & 2 & 3 & 4] \\ [5 & 6 & 7 & 8] \end{bmatrix}$$

2×4

$.\text{shape} = (2, 4)$

row_means.shape = (2,)

$$\begin{bmatrix} 2.5 \\ 6.5 \end{bmatrix}$$

(2,1)

$$\begin{bmatrix} 2.5 & 6.5 \end{bmatrix} \leftarrow \text{column\_axis}$$

(1,2)

(1,2)

(2,1) (2,)

(1,2), (2,1), (2,) → For a computer these 3 things are totally different

Rank of a Matrix according to Computer →

Number of dimensions

A → 2 ((2) and (4))

$\boxed{\text{row\_means} \rightarrow 1 \quad (2,1)}$ → Matrix multiplication on Matrix Addition or

Make sure that you convert numpy array of rank 1 or single dimension to rank 2 numpy array

(2,) → (1, 2) → [[ ]] → [2.5 6.5] X

[ ]

(2,) → (2, 1) → [[2.5] [6.5]] → $\begin{bmatrix} 2.5 \\ 6.5 \end{bmatrix}$

1 DIM ARRAY → VECTOR

VECTORS

MATRIX → 2-DIM ARRAY

COLUMN VECTOR

ROW VECTOR

0-DIM ARRAY → SCALAR .shape = ( )

0-DIM ARRAY $\longrightarrow$ SCALAR. Shape = ( )

1-DIM ARRAY $\longrightarrow$ VECTOR. Shape = (*,)

2-DIM ARRAY $\longrightarrow$ MATRIX. Shape = (*, *)

3 OR MORE THAN 3-DIM ARRAY $\longrightarrow$ TENSOR. Shape = (*, *, *)

= (*, *, *, *)

= (*, *, *, *, *)

5-DIM TENSOR

RANK IS THE NUMBER OF LINEARLY INDEPENDENT VECTORS IN A <u>MATRIX</u> → LINEAR ALGEBRA DEFINITION

NUMPY DEFINITION → NUMBER OF DIMS → ( ) → 0 SCALAR

→ (*,) → 1 VECTOR

→ (*, *) → 2 MATRIX

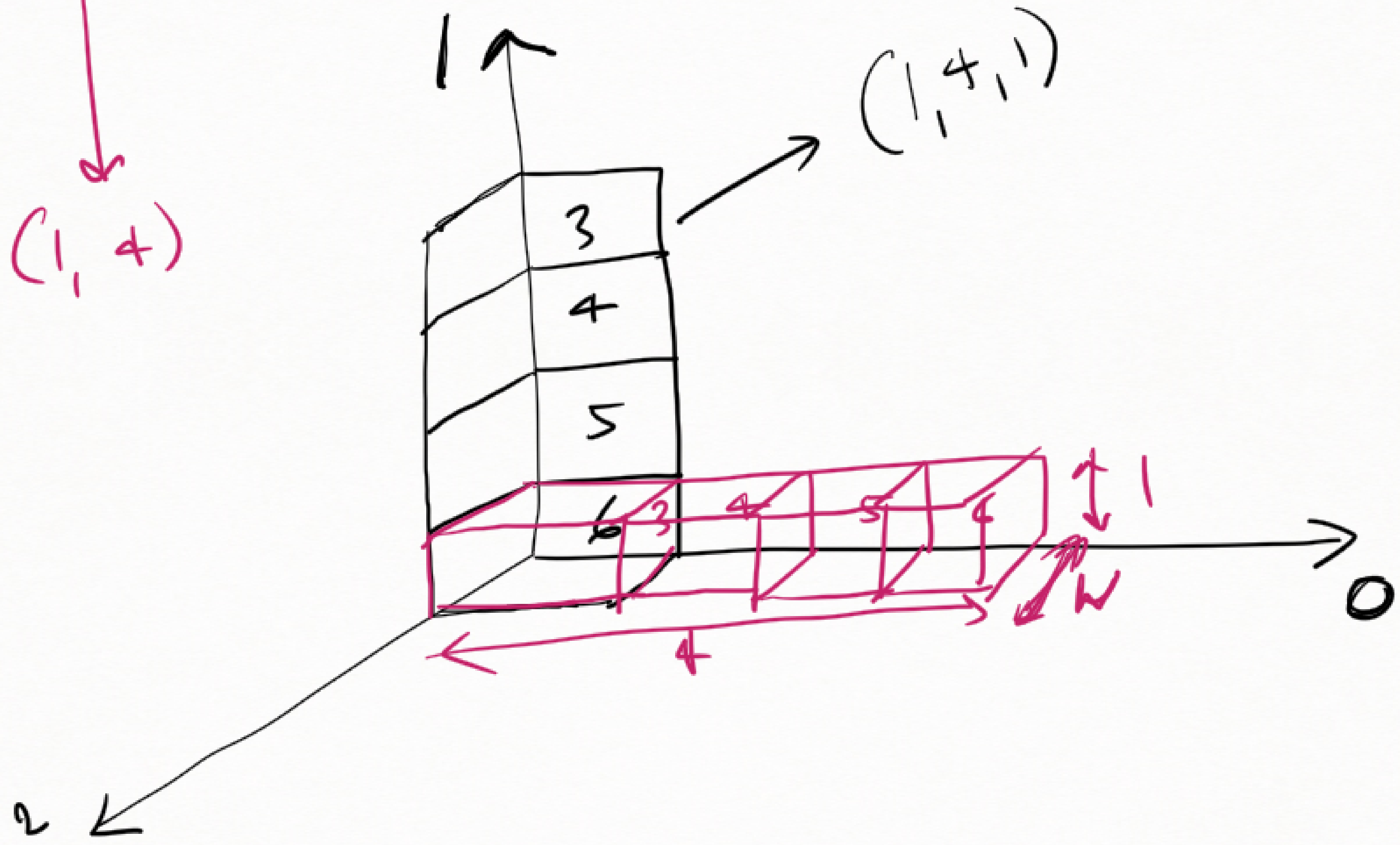
→ (*, *, *) → 3 <u>TENSOR</u>

$[3, 4, 5, 6] \rightarrow$ 1D VECTOR

$(4, 1)$

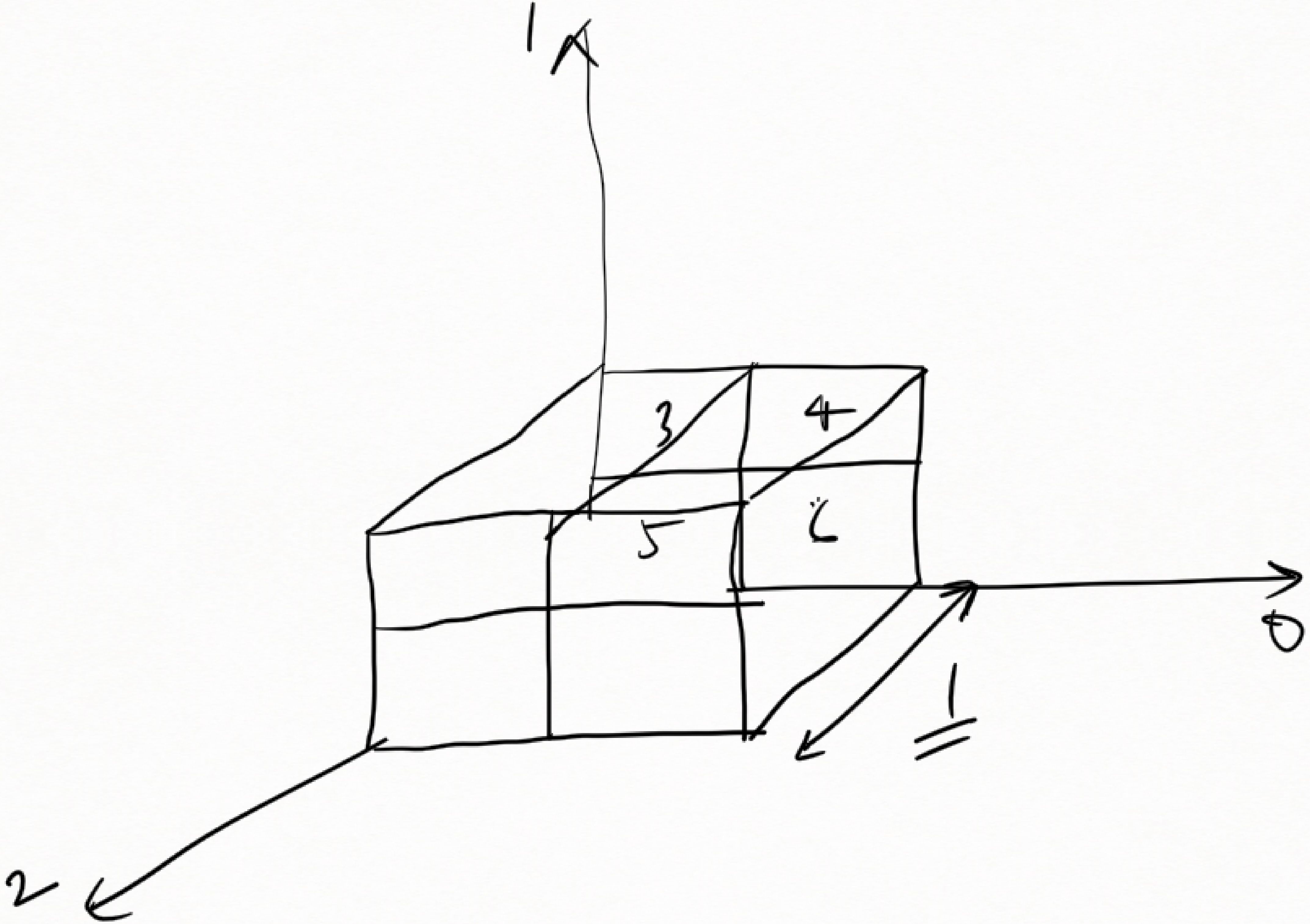
$(1, 4)$

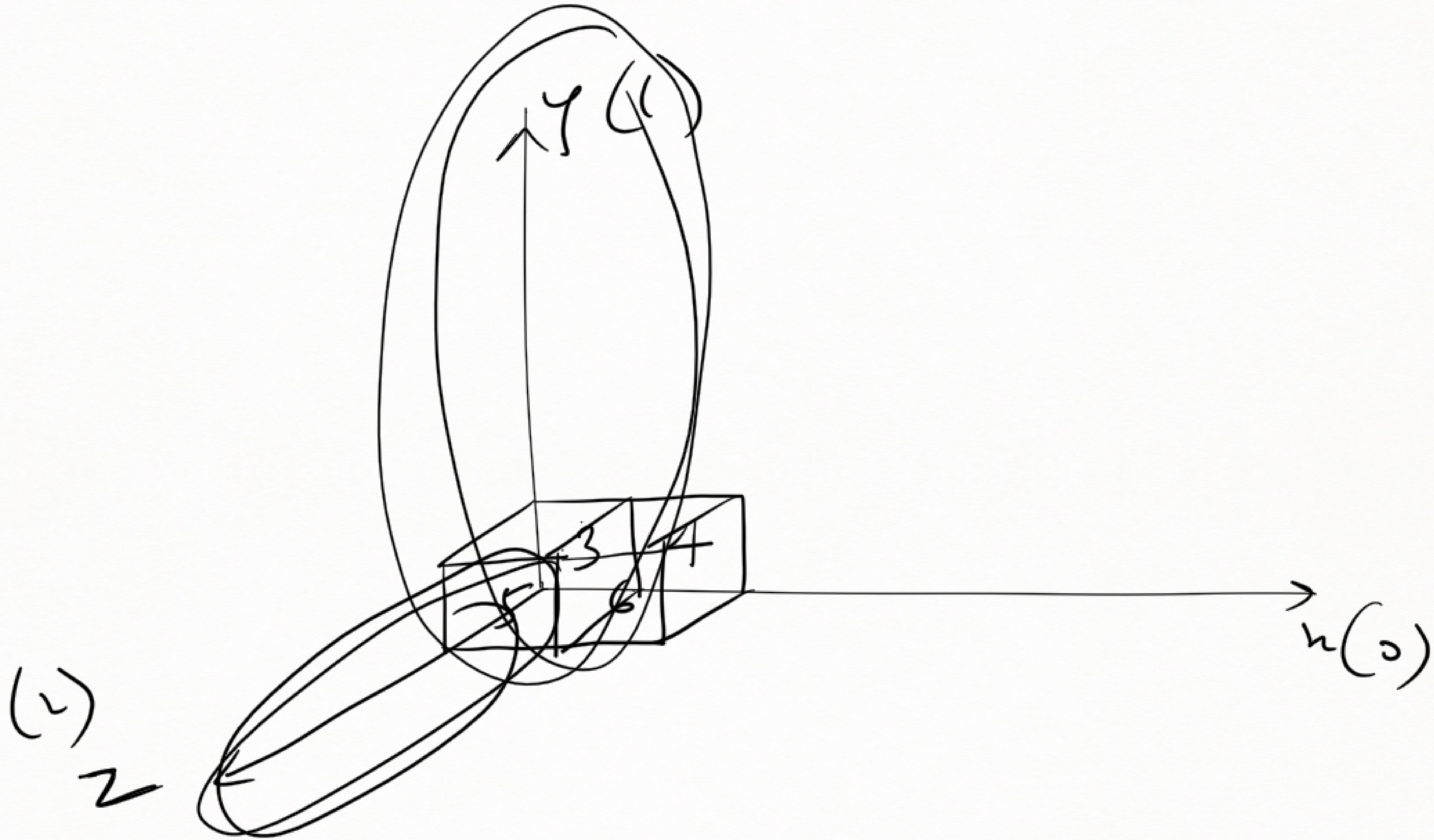
$(1, 1, 4)$



$(1, 2, \underline{\underline{1, 2}})$

$(2, \underline{\underline{1, 2}})$




Y (1)
x (0)
(2)
z
3
4
5
6

$(10, 5) \longrightarrow \underbrace{(50, 1), \ (1, 50), \ (25, 2), \ (2, 25)}, \ (50, 1)$

50   50   50   50   50

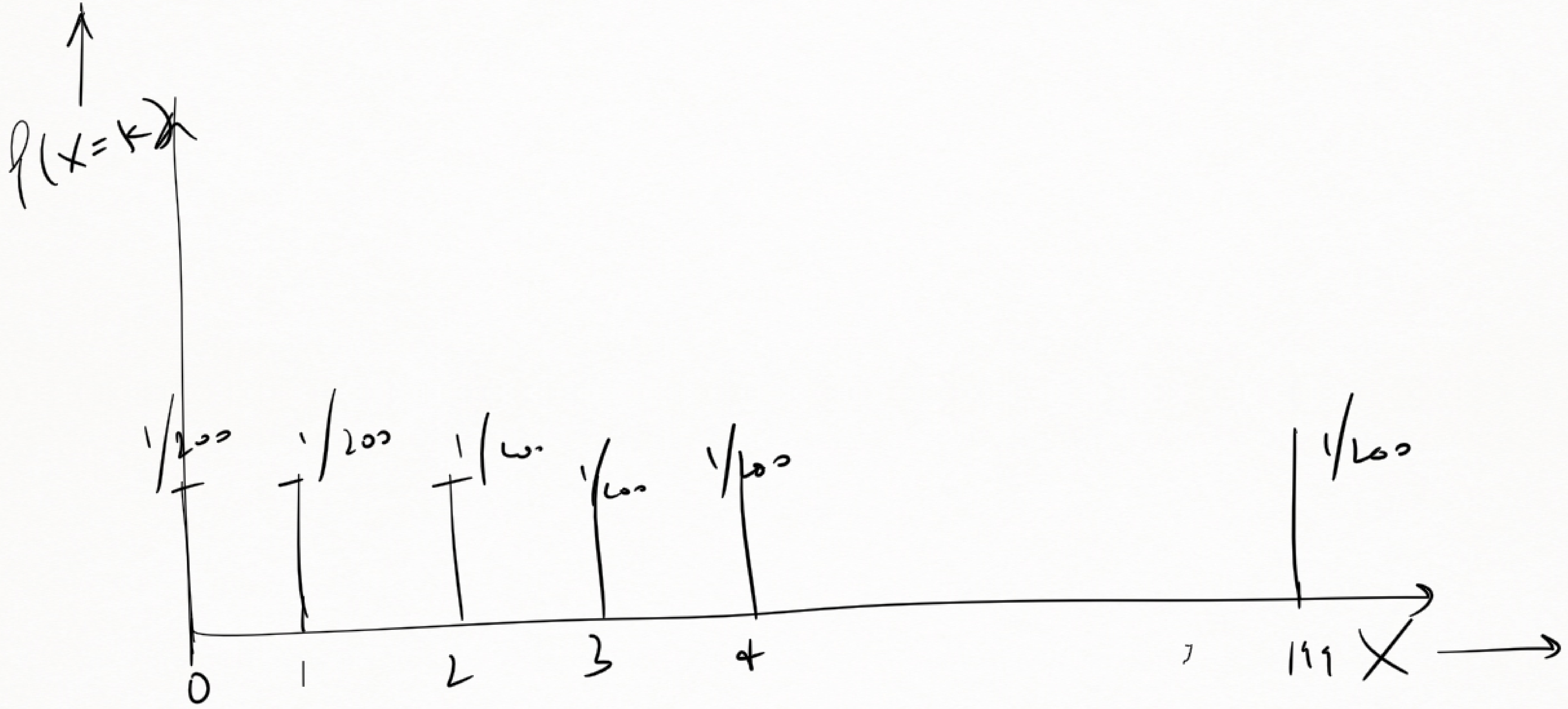
$$\begin{array}{cc} & \begin{array}{cccc} 0 & 1 & 2 & 3 \end{array} \\ \begin{array}{c} 0 \\ 1 \end{array} & \text{argmax}\left(\begin{bmatrix} [1, & 2, & 3, & 4], \\ [5, & 6, & 7, & 8] \end{bmatrix} \right) \end{array}$$

0, 1, 2, 3

axis = 0

1, 1, 1, 1

3, 3



p(X=k)
1/200
1/200
1/200
1/200
1/200
1/200
0
1
2
3
4
199
X

$$\begin{array}{ccccc} 0 & 1 & 2 & 3 & 4 \\ [114, & 156, & 139, & 114, & 26] \end{array} \longleftarrow \begin{array}{ccccc} \boxed{4 & 0 & 3 & 2 & 1} \\ 26, & 114, & 114, & 139, & 156 \end{array}$$

$$\underbrace{p_1 = 1/6, \quad p_2 = 1/6, \quad p_3 = 1/6, \quad p_4 = 1/6, \quad p_5 = 1/6, \quad p_6 = 1/6}_{\text{pvals}}$$

4 outcomes

$$\begin{array}{cccc} 0 & 1 & 2 & 3 \end{array}$$

$$pvals = [0.45, 0.3, 0.35, 0.10] = 1$$

$n = 1$ You are throwing a die single time

$n = 6$

$10 \times 4 = 40 \times 6$

$= 240$

(1000, 1), (1, 1000), (2, 500),

(500, 2), (4, 250), (250, 4), (125, 8)

(8, 125), (2, 2, 250), (2, 250, 2), (250, 2, 2)

(2, 4, 125), (4, 2, 125), (125, 4, 2), (125, 2, 4)

500 − 45

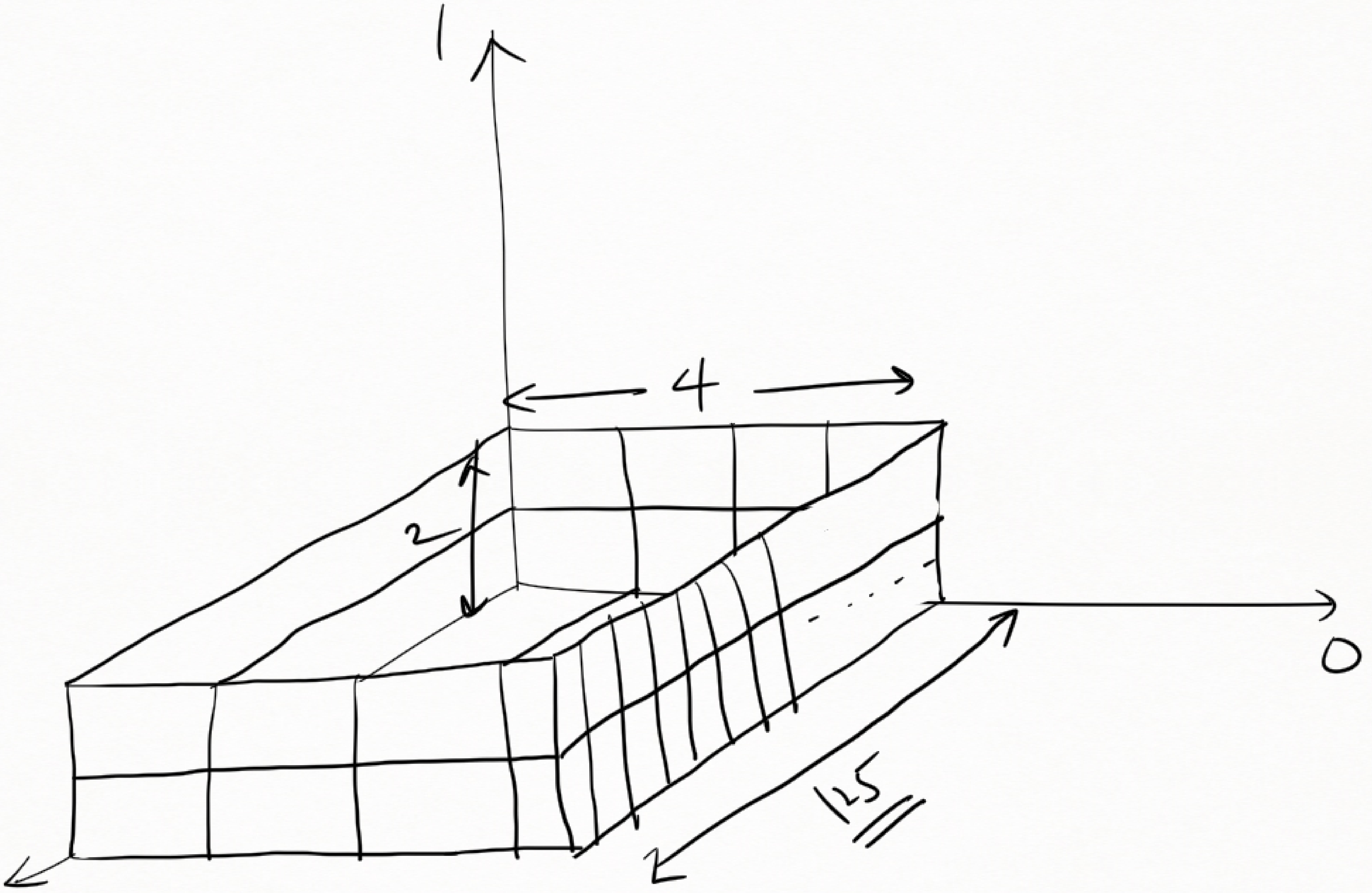
455

1/455 1/455 1/458 1/455

45 46 47 48 500 X

$$\underline{\underline{(15,\ -1,\ -1)}}$$

$$\left\{\begin{matrix} 4, & 2 \\ 2, & 4 \\ 1, & 8 \\ 8, & 1 \end{matrix}\right\} \begin{matrix} = 1000 \\ = 1000 \\ = 100 \\ = \underline{\underline{1000}} \end{matrix}$$



1
4
2
0
125