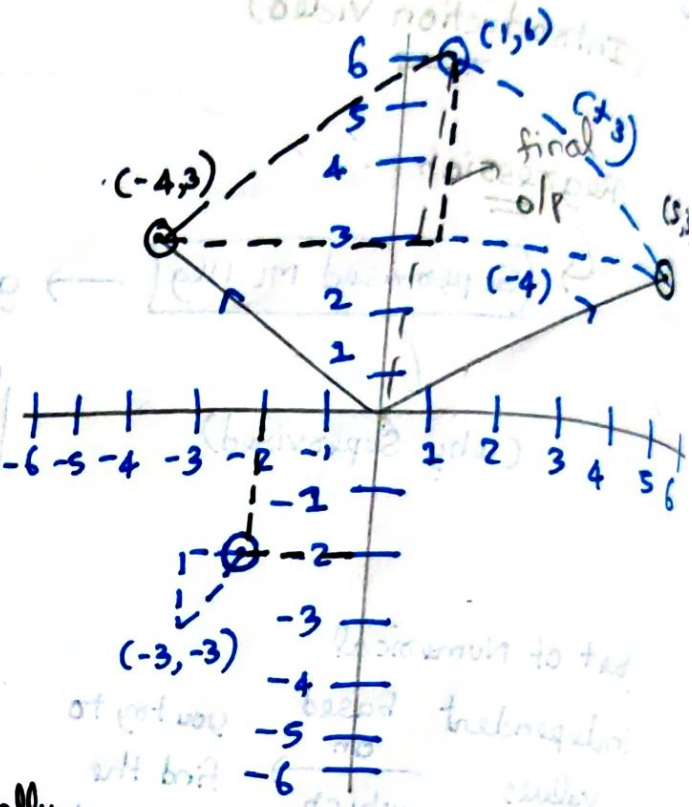
4. Addition of Vectors

$$P_1 = \begin{bmatrix} -4 \\ 3 \end{bmatrix} \qquad P_2 = \begin{bmatrix} 5 \\ 3 \end{bmatrix}$$

$$P_1 + P_2 = \begin{bmatrix} -4 \\ 3 \end{bmatrix} + \begin{bmatrix} 5 \\ 3 \end{bmatrix}$$

$$= \begin{bmatrix} 1 \\ 6 \end{bmatrix}$$



So basically what happened is (from the point (-4,3) we have moved 5 positions to the right) $\longrightarrow -4 + 5 \rightarrow$ 1 (right)

$$A = \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} \qquad B = \begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix}$$

$$A + B = \begin{bmatrix} x_1 + x_2 \\ y_1 + y_2 \\ z_1 + z_2 \end{bmatrix} = \begin{bmatrix} x_3 \\ y_3 \\ z_3 \end{bmatrix}$$

Another Example

$$A = \begin{bmatrix} -2 \\ -2 \end{bmatrix} \qquad B = \begin{bmatrix} -1 \\ -1 \end{bmatrix}$$

$$\text{ans} \rightarrow \begin{bmatrix} -3 \\ -3 \end{bmatrix}$$

Example (w.r.t Data Science)

Solving a use case

Sensor 1 [3, 5, 7]

Sensor 2 [2, 4, 6]

↓ In Data Science we use this for

① Data Aggregation Tasks

② feature Engineering

EDA
↳ Exploratory Data Analysis

FE
↳ feature Engineering

↓

final Sensor Reading

Sensor (1+2)
↳ [5, 9, 13]

Data Science
↳ we work in a field called
NLP

we have an E-commerce website where people put reviews of specific product

→ where you have

Reviews	Sentiment
product is good	1
product is bad	0

If so u are training a model for this

I/p → Model (String) → Sentiment

↳ These i/p's which we feed to the model gets converted to vectors

→ a lot of math calc that happens inside the model (it can't understand strings)

for converting

Text → vector (we use diff embedding techniques)

→ ① One hot Encoding

② TF → Term freq

IDF ↳ Inverse Doc freq

③ CBOW

④ word2vec

Every Sentence is converted to vectors

[Numerical values inside]

[- - - - -]

↓

Numerical values

Eg

word Embeddings

Data → [0.2, 0.1, 0.4]

Science → [0.3, 0.7, 0.2]

Data Science

↓

$V_{Data} + V_{Science}$ →

[0.2, 0.1, 0.4]

+ [0.3, 0.7, 0.2]

= [0.5, 0.8, 0.6]

Data Science (vector)

③ Image Processing

A colour Image will be represented as → [R G B]

① Red channel → R → [255, 128, 0]

② GREEN channel → G → [128, 255, 0]

③ BLUE channel → B → [64, 64, 255]

In order to convert RGB → Gray Scale → $\frac{[255+128+64,\ 128+255+64,\ 0+0+0]}{3}$

for every cell

So, we get

[149, 149, 85]

final white and black