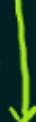


**Assignment:4**

1. you have to create one supervisor node.
2. create one router function
3. create three more noe
- 3.1 llm call (llm node)
- 3.2 RAG (rag node)
- 3.3 web crawler(fetch the info in realtime from internet)
4. created one more node after this for validation for generated output --> explore the validation part how to do that
5. if validation going to be failed in that case again go to supervioser node and then supervisor node will again decide what needs to be call next
6. once the validation will pass then only generate the final output

result["merge"][-1] =

{"merge": ["what is GDP of USA"]}



Supervisor

↳ question → state["merge"][-1]

---

Topic → "USA"



state = {"merge": ["what is GDP of USA", "USA"]}

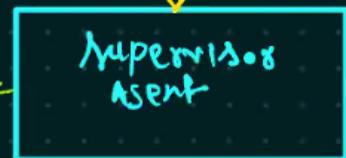
class AgentState(typecheck):

Input question user (hi, hello, ...)

class Topic ---

↑ Validation

{"messages": [ ]} → state



{"messages": [ "hi" ]}

↓  
Pydantic output form  
{}

< state["message"][-1] > → **Hi** { Topic = "NOT Related" } → Parser  
renaming = "not related"

state = {"messages": [  
    "Hi",  
    "Not related"]}

LSTM cell

< code > ← final answer

state = { "messages": [ "hi", "Not related" ] }  
      |  
      |  
      0           -1

state = { "messages": [ "Hi",  
"Not related" ] }

< Router >

state["messages"][-1] = "not related"

LLM call

< code > ← final answer

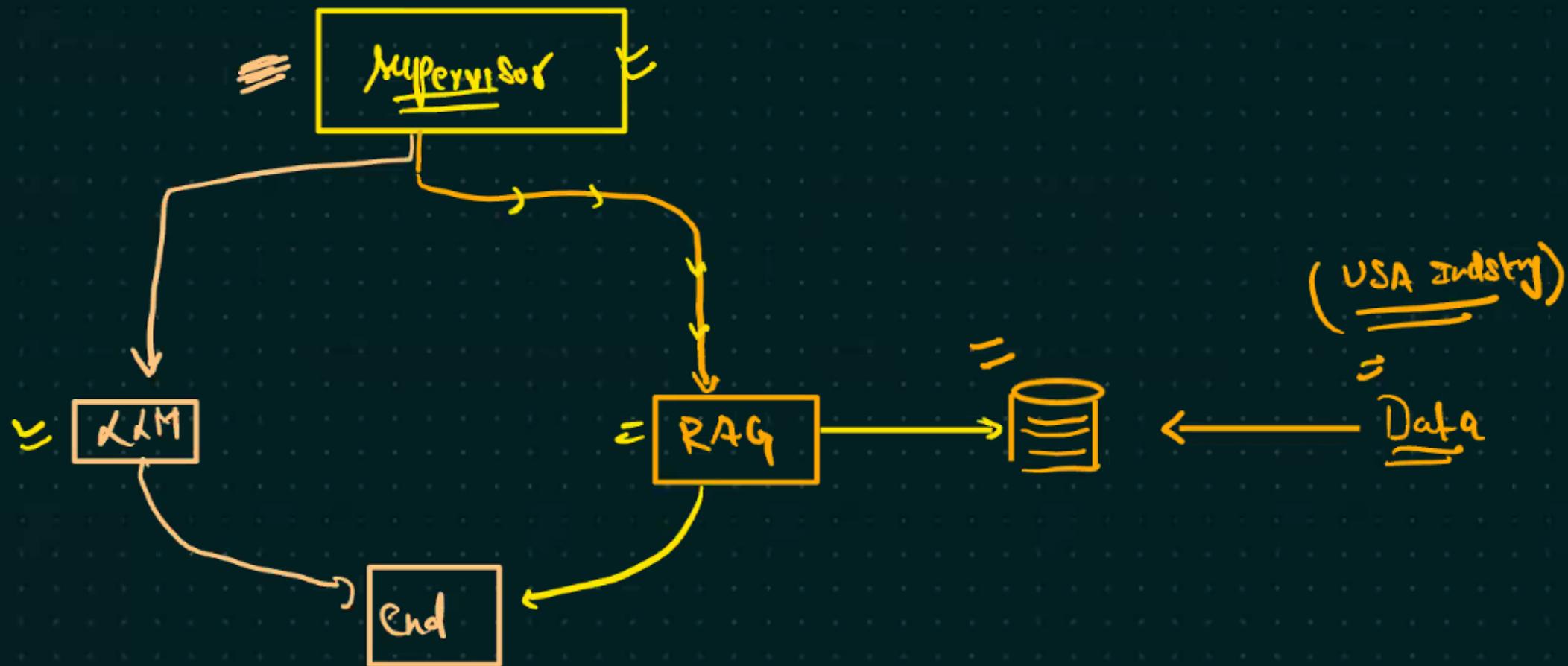
state = { "messages": [ "Hi", "Not related" ] }

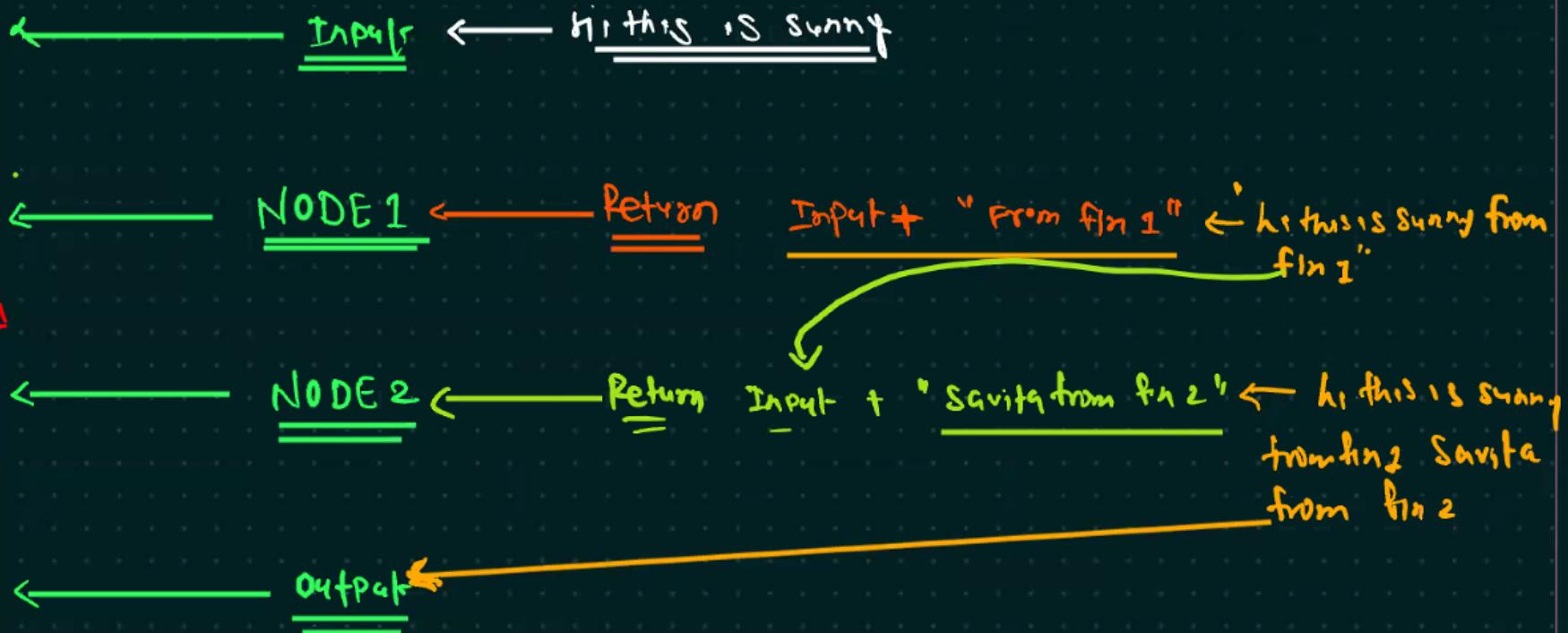
< state["messages"][-1] > → Hi  
TOPIC = "Not Related"  
renaming = " "

Parser

Start

Hi, how are you? | What is current GDP of USA?





NODE + EDGE

Lang graph

NODE + EDGE

Functions

Relation b/w those fn

Langraph  
Ageric flow

Input

Acyclic

$\Rightarrow$  Graph

Langraph  
Node + edges

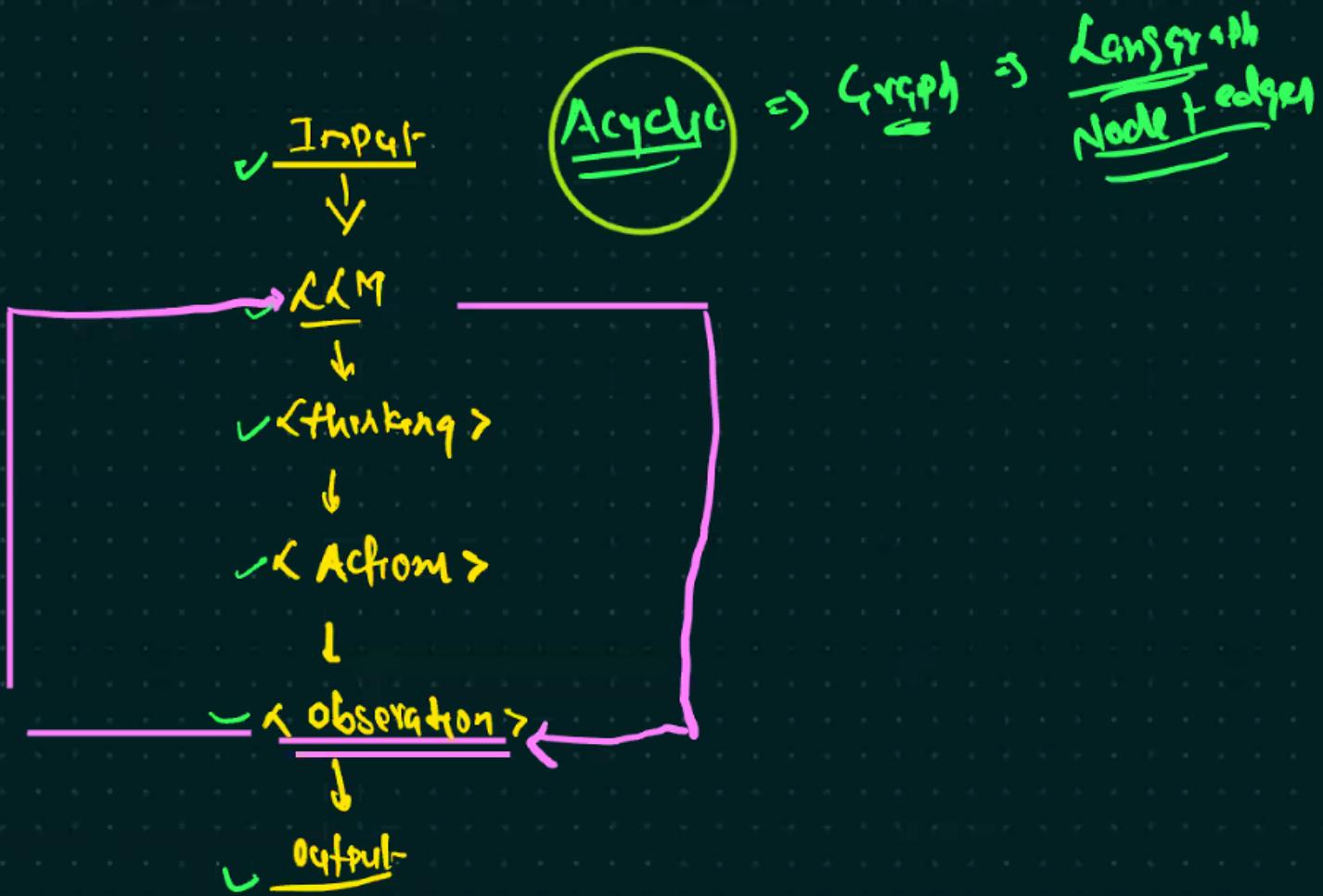
# Languraph

## Agentic flow

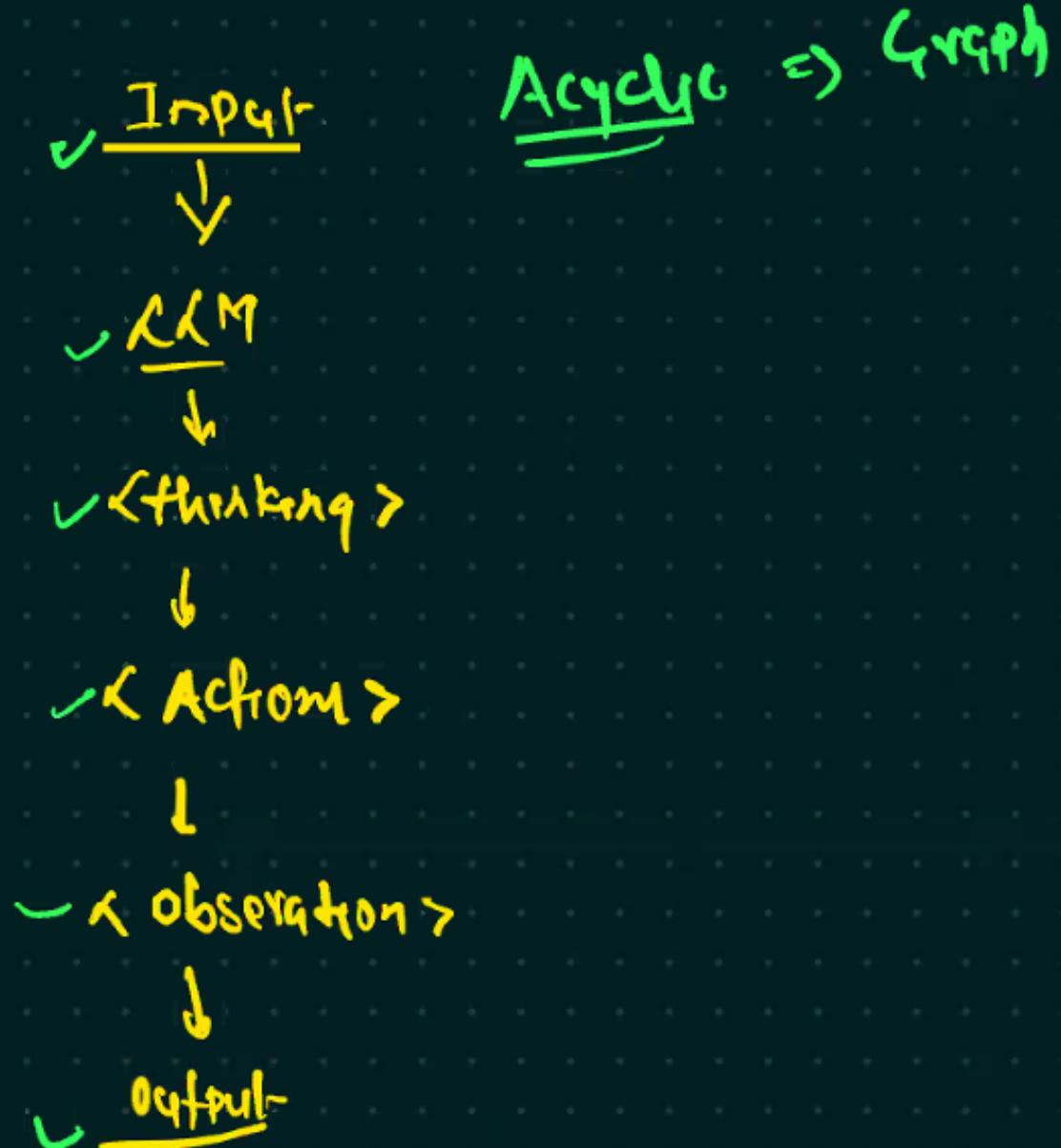


FUNCTIONS

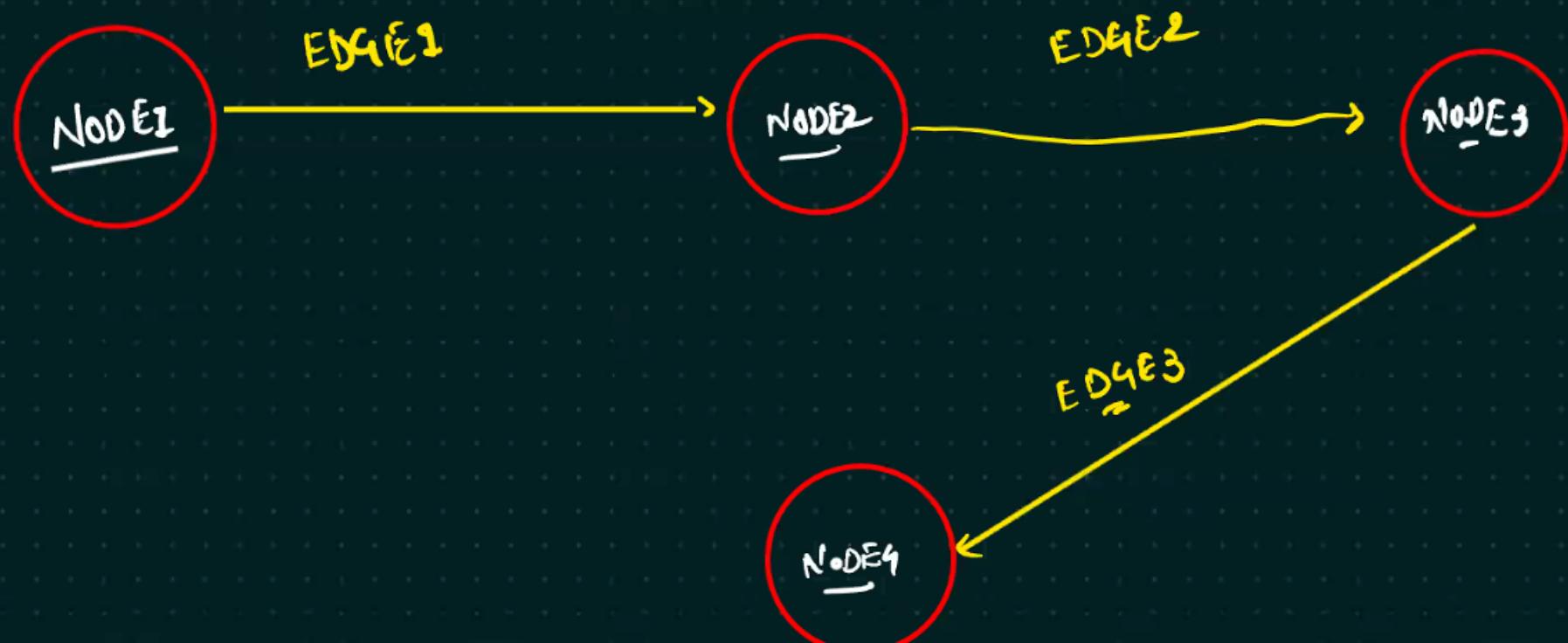
FUNCTIONS



## Agentic Flow



Graph  $\rightarrow$  Ayclic graph



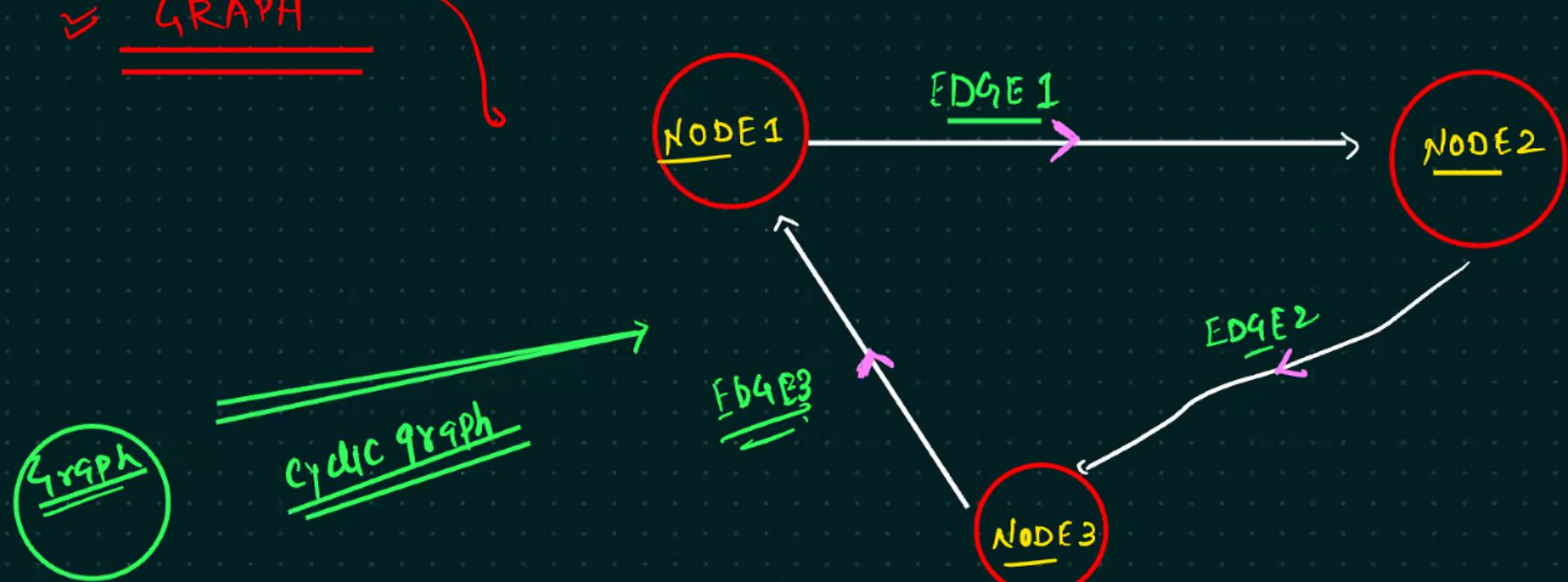
Datastructure

(Functions)

Connectivity between those functions

Relationship b/w the fm

GRAPH



question → What was the stock price of APPLE last year?

Ans → APPLE → \$\$\$

question - Can you tell me the today's opening price of APPLE?

Ans → think (today's opening price) ×

→ Action (tool calling) → = Google search API

Google search API

button

custom tool

Agent  $\Rightarrow$  ALM + tool

$\Rightarrow$  Langraph  $\Rightarrow$  Orchestrator framework  $\Rightarrow$  Agent workflow

$\downarrow$   
 $\Rightarrow$  Workflow

ALM - think, Action, Observation

Planning

human capability

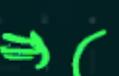
brain

Action



Langchain  
inbuilt tool  
custom tool

= Langgraph



{ Graph

Node  
edges

stateless → state  
conditional edges

Orchestrator  
Workflow



