→ Motivational Info.

→ Theoritical Definition about AI

↳ Mimic Human Intelligence in Machine (computing machine)

(Rationale Agent.)

↳ Agent (Program, Hardware Module, Obj)

↳ Act like humans

Ex: Human, Robot, Vacuum Cle

↳ Autonomous Driving

↳ Smart system. Alexa...

↳ Decision Making (Computation | Pred)

SAI Module
Framework

↳ Environment

↳ Static (Rooms, Board)

↳ Dynamic (Autonom. Keeps on Changing.

↳ Fully observable.

Input,
Action,
output / Display

(You observe all the possible states)

Partially observable

function output → Sensors (Receive the data.)

IoT devices.

Actuators | To act / Pen

Game play

Problem A → M_1, M_2, M_3 . . . . . M_k Performance metric

Price
Quality
Industries...

Time to finish the game
Number of steps taken to win the game

Classification : Accuracy: How many sample the model correctly classifys
: Precision, Recall, F1-measure
Harmonic Mean Pre

Clustering : Inter & Intra Cluster distances

Regression: score as the output
Mean Squared Error
MAbsolute Error
Mean Absolute Percentage Error.

Translation (Sourcee language to Target language)

Source: राम एक अच्छा लड़का है। (Hindi)
Target: Ram is a good boy.

Performance Metric:
BLEU Grammar (linguistically Correct)
ANGH How many words in target language are

KUV, {correctly "translated."

→ Dashboard (car) Camera : Recording.

↳ $C_1, C_2 \ldots C_k$

Performance metric

→ Resolution; FPS, NOISE.

High speed & Low speed {Quality of the Img}

Response time: High Temp; Low Temp

→ Conversation AI (Alexa):

↳ $A_1, A_2, \ldots A_k$

↳ Speech to Text (How good is it?)

↳ # time it doesn't respond that is expected.

↳ Time to respond. ( )

↳ Under Noise

→ Optimization: Convergence speed

optimum soln quality

minimum / maximum } ↳ $O_1, O_2 \ldots O_k$

~~Ex.~~ time, space, ~~Quall~~, Completeness.

Water Jug. (3L, 5L) → 2L, 1L, ~~7~~L ....

River Crossing. (Man, food, Cabbage, Lion} boat)

A I Module:

- Performance Measure
- Environment
- Actuators
- Sensors

PEAS

Percept

meter to cm

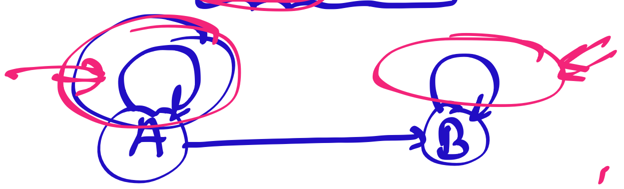
→ **Goal Based Problems** { min break }

↳ Given a Problem, model it as a graph problem:

Graph : Set of Nodes (Vertices) / States { = Empty

Set of Edges (pair of nodes) = {φ, ...

Simple Graph (without self loops)



$G : (V, E)$

$V = \{ v_1, \ldots \ldots \}$

$E = \{ e, \ldots \ldots \}$

Grap

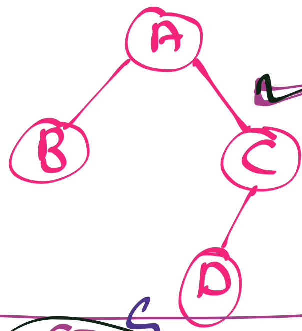
A

$V = \{A\}$

$E = \{\ \}$

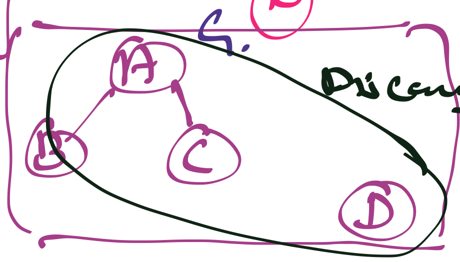
Graph ✓

Connected Acyclic graph. } → Tree ✓



Tree?

Tree : Graph Acyclic + Connected



Disconnect

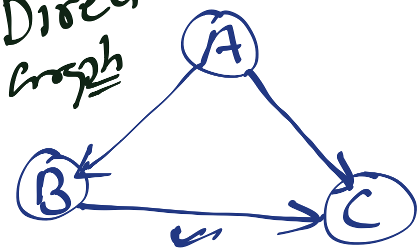
Is it a Tree ? NO

→ Types Graph:

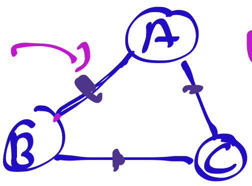
↳ Directed (→)

↳ Undirected. (⇄)

Directed Graph



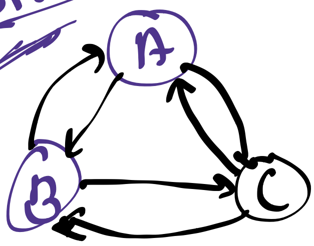
Undirec Gr.



✓ G ({A, B, C}, {(A,B), (B, C), (A, C)})

G ({A, B, C}, {(A,B), (B,C), (C,A)})

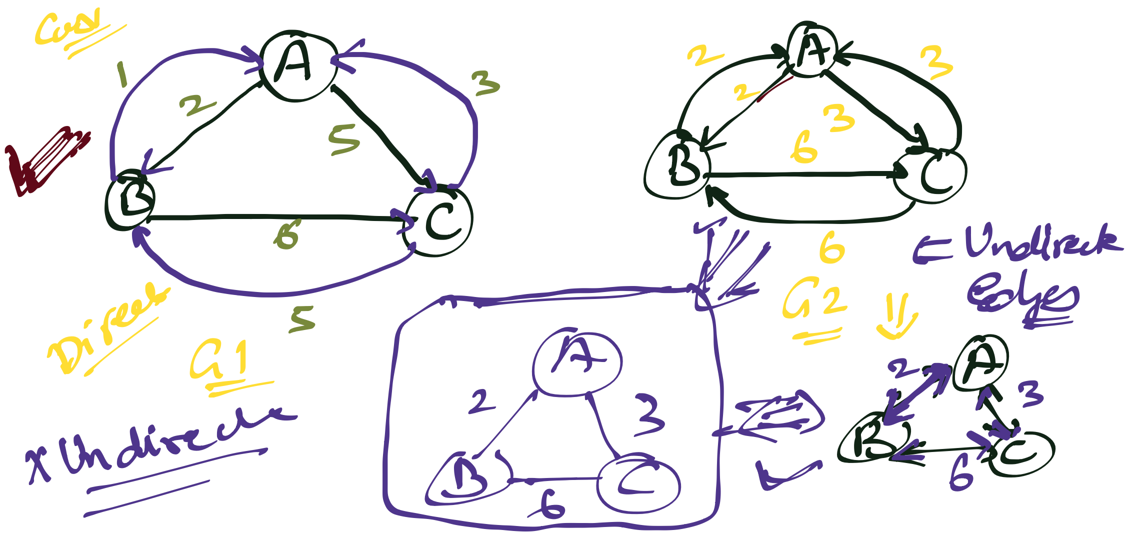
Directed



$$\Rightarrow E = \{ (A,B), (B,A), (A,C), (C,A), (B,C), (C,B) \}$$

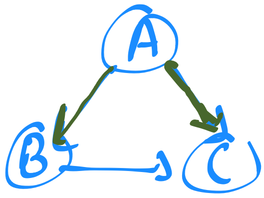
$$\boxed{(v_i, v_j), (v_j, v_i)}$$

Bi-direction



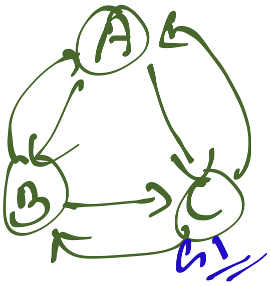
Degree of a Node (Vertex / state) in a graph G.

⇒ The total Number of incoming Outgoing Edges (links) defines the degree of the node.



$deg(A) = In(A) + out(A)$
$= 0 + 2 = 2$

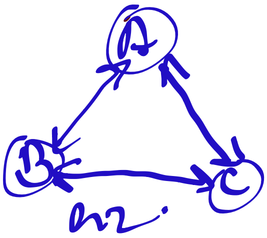
$deg(C) = In(C) + out(C) = 2$
$= 2 + 0$



$deg(A) = In(A) + out(A)$
$= 2 + 2 = 4$

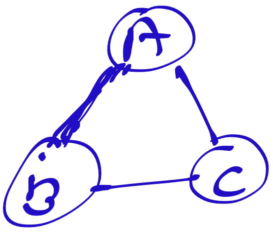
$dge(B) = In(B) + out(B)$
$= 2 + 2 = 4$

$dg(C) = Ing(C) + out(C)$
$= 2 + 2 = 4$



$dg(A) = In(A) + out(A)$
$= 2 + 2 = 4$

$dg(B) = 4$, $deg(C) = 4.$



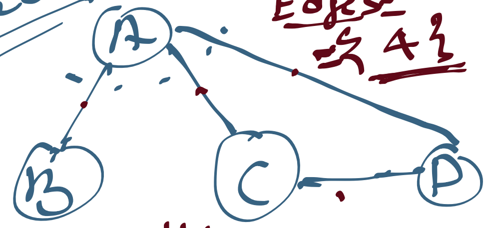
$\Rightarrow$

$deg(A) = In(A) + out(A)$
$= 2 + 2 = 4$

$deg(B) = In(A) + out(B)$
$= 2 + 2 = 4$

$deg(C) = In(C) + out(C)$
$2 + 2$
$= 4$

Undirected



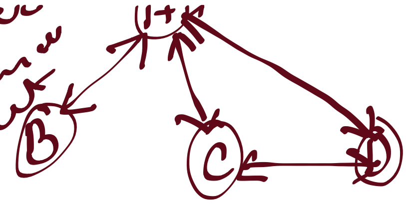
Edges = {4}

$deg(A) = \underbrace{In(A)}+out(A)$
$= 3 + 3 = 6$

$deg(B) = In(A) + out(B)$
$1 + 1$

$deg(C) = In(c) + out$
$2 + 2 = 4$

directed
separate
undirected



$d_T(D) = In(D) + out(D)$
$= 2 + 2$
$= 4$

→ Undirected (Similarly you can go for Directed)

Fully Connected graph

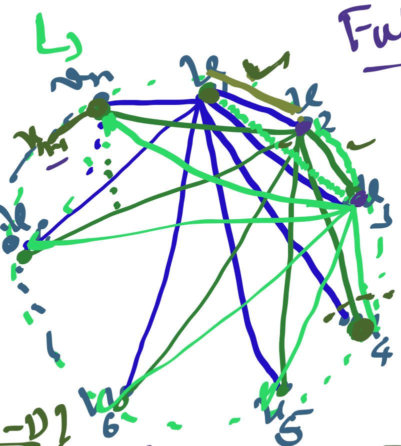
$G: V = \{v_1, v_2, \dots v_n\}$

Every node is connected to every other node

How many Edge

$\therefore n * (n-1)$

$\Rightarrow (n-1) + (n-2) + (n-3) + \dots 1$



$^nC_2$

$= \left\{\frac{n \cdot (n-1)}{2}\right\}$

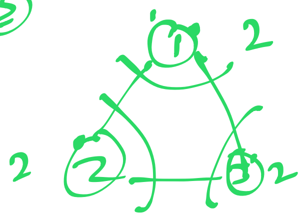
→ Fully Connected Undirected ($v_i \to v_j$)

⇒ ($v_j \to v_i$)

$^nC_2 = \frac{n!}{2!(n-2)!} = \frac{n(n-1)}{2}$

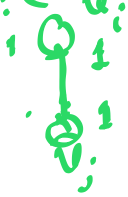
$Deg(FCG)^{in/out} = (^nC_2) \times 2$

$^3C_2 = \frac{3!}{2!\,1!} = 3$



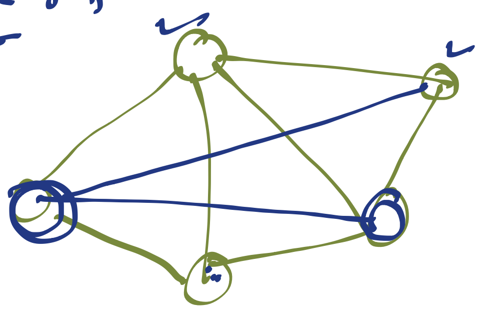
$\Rightarrow$ 6 (in) + 6 (out)

= 12



Partially Complete Graph (Most of of node/state are connected with the every other nodes

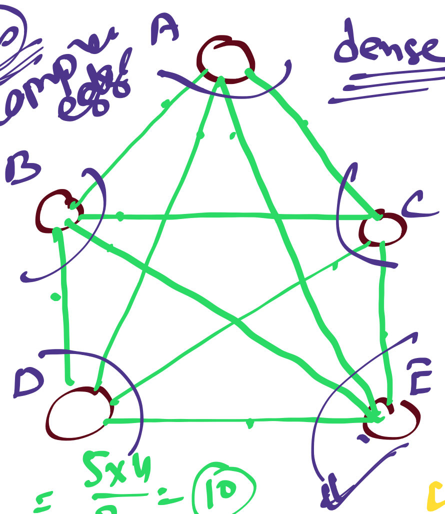
In. Complete graph



60% of the links are not ~~there~~ in the graph

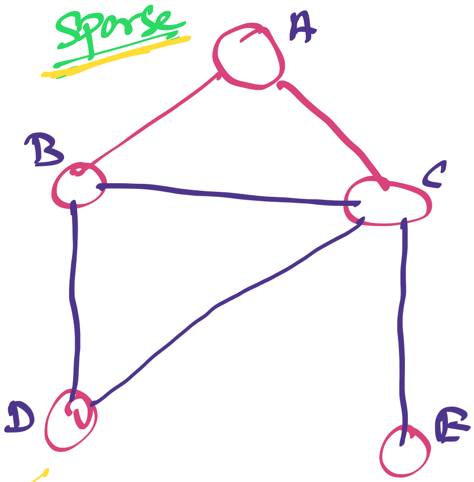
90%
A complete graph

dense



$$^{5}C_2 = \frac{5 \times 4}{2} = 10$$

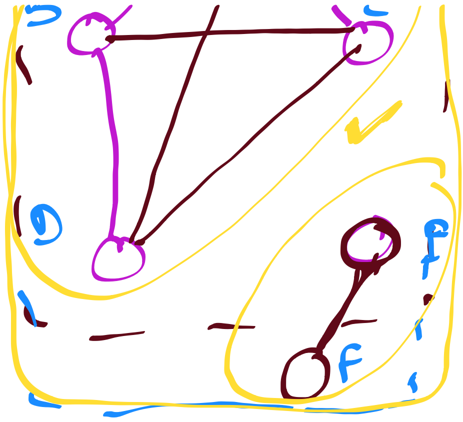
sparse



Connected? Disconnected?

$G = \{A, B, C, D, E, F\}$

$E = \{(A, B), (A, C), (B, C)$



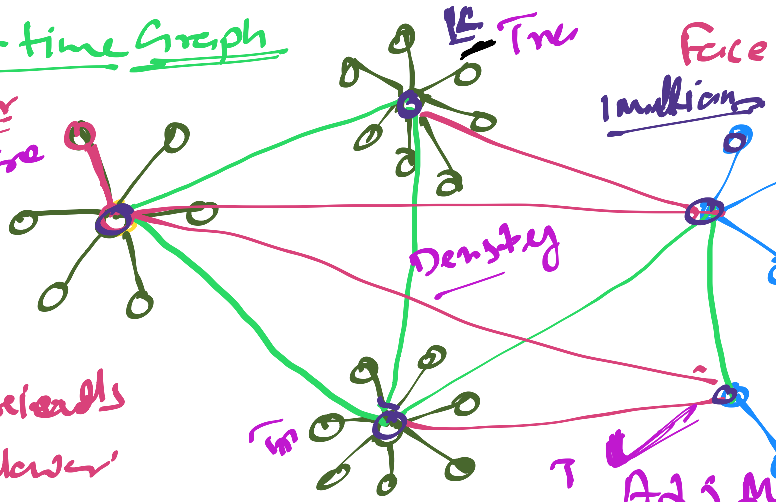
{(B,D),(C,D),(E,F)}

Disconneet

Real-time Graph

User

Free



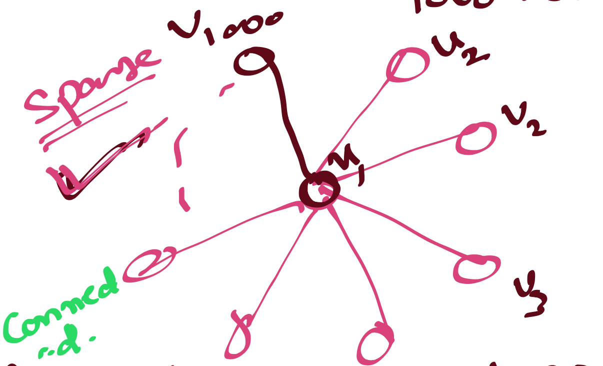
Facebook

Social Media

User & friends followers

Adj Mat & Adj List ? ?

1000 node



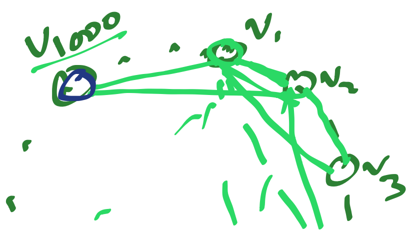
$G = \{v_1, v_2, \ldots v_{1000}\}$

$E = \{(v_1, v_2), (v_1, v_3), (v_1, v_4) \ldots (v_1, v_{1000})\}$

1000 nodes
999 edges

Complete graph with 1000 nodes will have

$$^{1000}C_2 = \frac{1000 \times 999}{2} = 50k$$

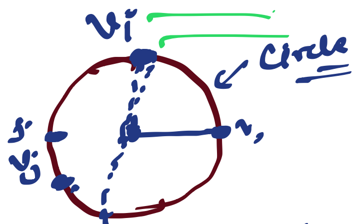


$$\frac{(999 \times 1000)}{2} = (\quad)$$

Diameter of a Graph

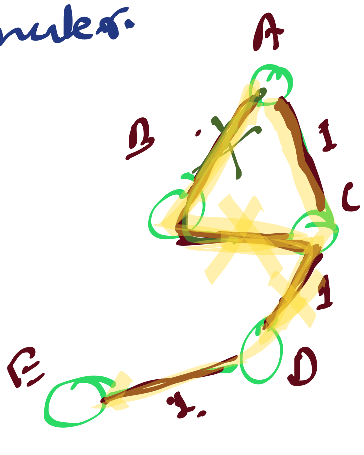
$= \max d(v_i, v_j)$

maximum distance b/w two points (v_i, v_j) on the circumference if circle circumfer..



The maximum Distance b/w a pair of nodes/states in the graph is called diameter.

G:



diameter(G) =

max dist{ (A,B), (A,C) A,D ...

$\underbrace{\text{dist}(A,E)}$ = 3 hops

max of the shortest dist

Adjacency Matrix

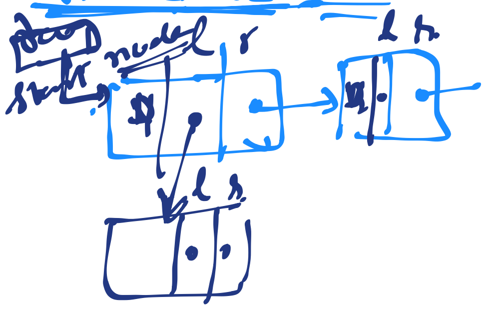
Stores the information of adjacent Nodes (Vertices) states

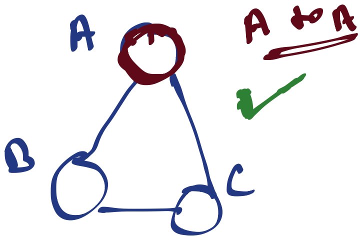
Array ✓ v_i — v_j

To Store edge / link Connection Info.

Adjacency List

Linked List DS





$N = 3$

Adjacency Matrix

	A	B	C
A	0	1	1
B	1	0	1
C	1	1	0

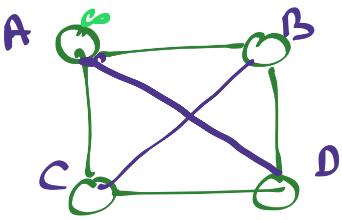
#nodes × #nodes

$(v_i, v_i) = 0$

$i \neq j \quad (v_i, v_j) = 1$

6 ed

G1:



12 Edges

$$\begin{array}{c|cccc} & A & B & C & D \\ \hline A & 0 & 1 & 1 & 1 \\ B & 1 & 0 & 1 & 1 \\ C & 1 & 1 & 0 & 1 \\ D & 1 & 1 & 1 & 0 \end{array}$$

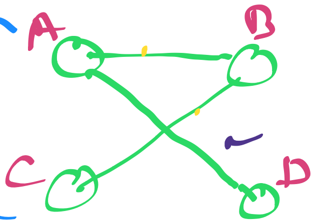
$v_i \to v_j$

$= v_j \to v_i$

symmetric

G2:

3 edges



boolean

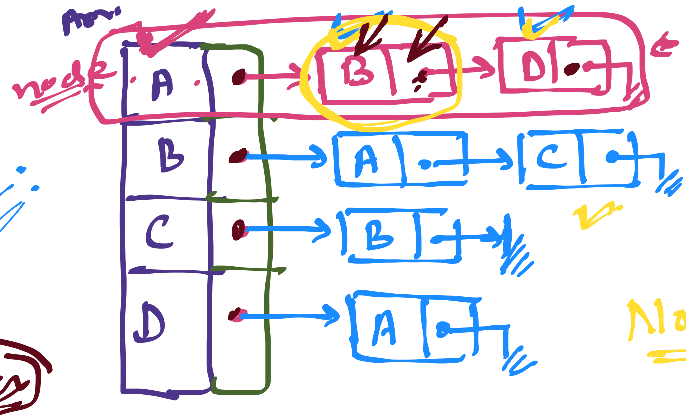
$$\begin{array}{c|cccc} & A & B & C & D \\ \hline A & 0 & 1 & 0 & 1 \\ B & 1 & 0 & 1 & 0 \\ C & 0 & 1 & 0 & 0 \\ D & 1 & 0 & 0 & 0 \end{array}$$

6 Zeros

lot of Zeros

6 one

G2:



List
Adjacent to nod A

More memo.

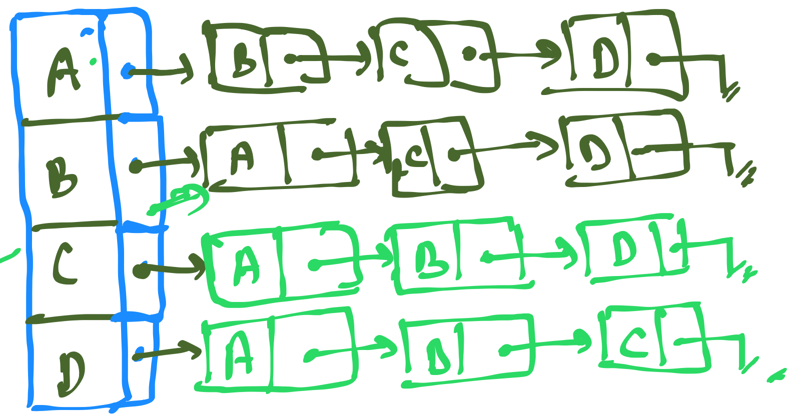
6 Edges

No Zero

4 bytes

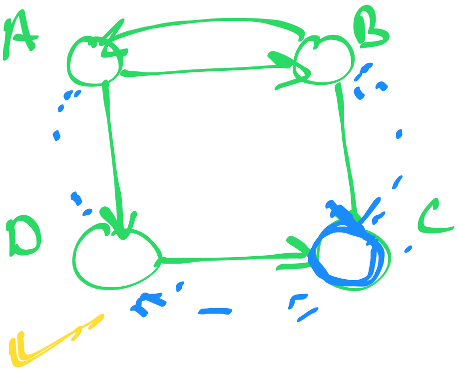
name of the node & pointer to other address of the linked node.

G1:



12 Edges

A → B
B → A



	A	B	C	D
A	0	1	0	1
B	1	0	1	0
C	0	0	0	0
D	0	0	1	0

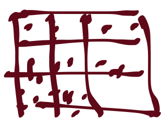
Not symmetric

n- nodes Undirected graph

Complete graph

$n \times n \times 1\,\text{bit} = n^2\,\text{bit}$

Adj Mat
boolean



Adj List

address required let K bits

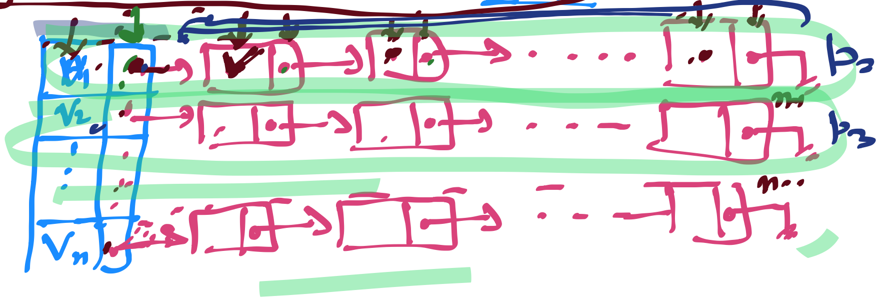
32 bits

1 bit

$n \times n$

p_1 neighbor

$n^2 \rightarrow$ address field



Number of bits required to store all addresses

Gu $n^2 \times K$. bits

4 bytes

$v_1 \;\; . \;\; . \;\; v_n$

Node is indexed = $\{0, \ldots, n-1\}$

$n = 6 \Rightarrow \{0, 1, 2, 3, 4, 5, 6\}$

00	0
01	1
10	2
11	3

000
001
010
011
100
...

$\lceil \log_2 n \rceil$

To store the node indexes: $\lceil \log_2 n \rceil$ Bits for one index will be required.

Total memory Needed $= (n \times \lceil \lg n \rceil) * n$

$= n^2 \cdot \lceil \lg_2 n \rceil$ bits

Total mem including add.

$= \boxed{(n^2 \cdot \log_2 n + n^2) \text{ bits}}$

Let p_i defines the number of adjacent nodes to i^{th} node.

Total number of Edges $= \left(\sum_{i=1}^{n} p_i + n \right)$

no. of edges

$\sum^{n}$

The memory for node str: $\left(\left(\sum_{i=1}^{n} D_i\right) + n\right) \lfloor \log_2 n \rfloor$

$+ \left(\left(\sum_{i=1}^{n} D_i\right) + n\right) \cdot k$ bits

$\sum_{i=1}^{n} D_i$ is very small in compare to n^2

it is sparse graph. For large 'n'. Adj list representation is beneficial

For Full connected graph

$$D_i = n-1$$

$$\sum_{i=1}^{n} D_i = (n-1)+(n-1)+\dots \quad n \text{ times}$$

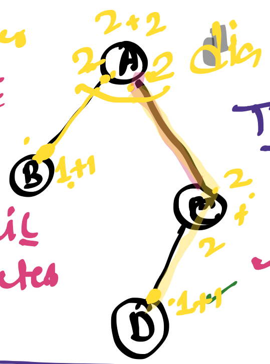
$$= n(n-1)$$

Adj Matrix	Adj List
✓ n^2 bit $[\ \ ]_{n*n}$	Let $\sum_{i=1}^{n} p_i$ be 'M'
boolean Matrix	$(M+n)\lceil \log_2 n \rceil$
Irrespective of number of edges.	$+(M+n).K$
	$= \boxed{(M+n)(K+\lceil \log_2 n \rceil)}$
←	let $M \le < n(n-1)$

is beneficial for large n, and small 'M' Adj list

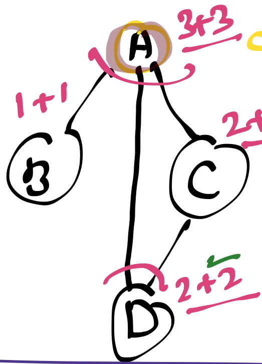
Total deg = 12

dia = 2



TREE

✓ Acyclic
✓ Connected



dia = 1

Total degree = 16

Cyclic
Connected
NOT A TREE

Connectedness	✓	✓
# nodes	4	4
# edges	3	4

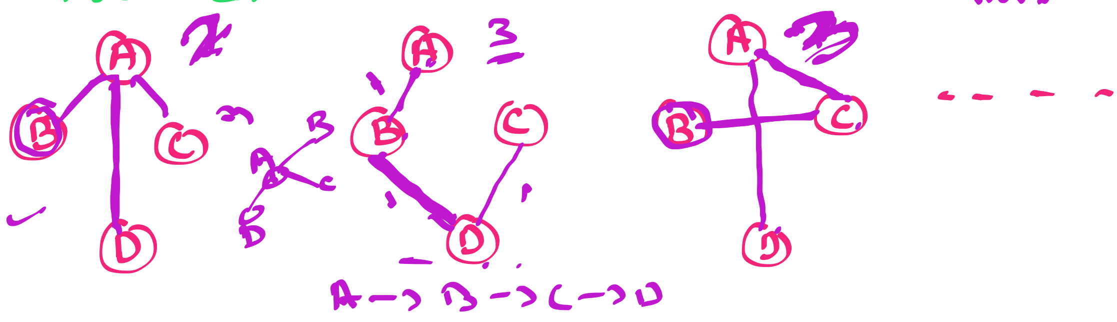
#cycles 0 | 1

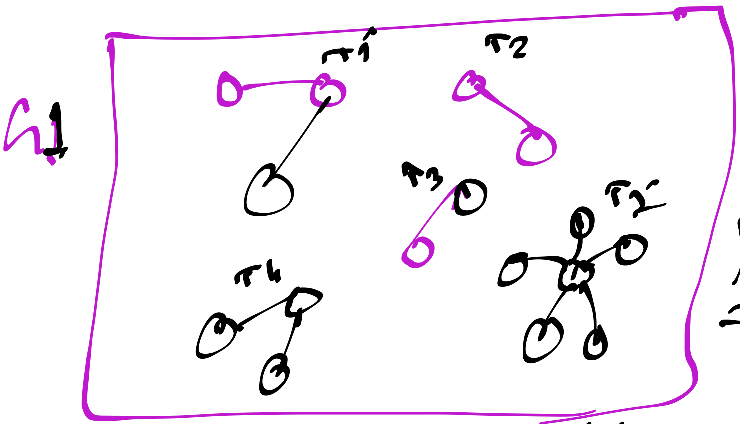
If a graph is Acyclic & Connected we call it tree. The number of Edges in the Tree Data Structure will be the (number of nodes - 1). If you add one - more edges in this DS it becomes cyclic & no more **TREE** IN our

Total Deg of TREE = 2 (numbenodes -1) + (number
= 2x (number of nodes -1) ... -1)

TREE (root Info) → First node we try to access

Maximum Diameter Undirected Unweighted Tree can have is (number of nodes −1) mins.





Collection of Trees

FOREST

DS ? Adj Matrix & Adj List