Graph Theory

Tree & Rooted Tree

Graph Theory :

Recap :

Complete Bipartite Graph

Website : https://www.goclasses.in/

Definition 62. A bipartite graph $G = (A, B, E)$ is *complete bipartite* if every vertex of A is adjacent to every vertex of B. A compete bipartite graph with parts of size n and m is denoted $K_{n,m}$.

$G(V, E)$ ✓

In Bip Graph

$V = \underbrace{A \cup B}$

$G(A, B, E)$

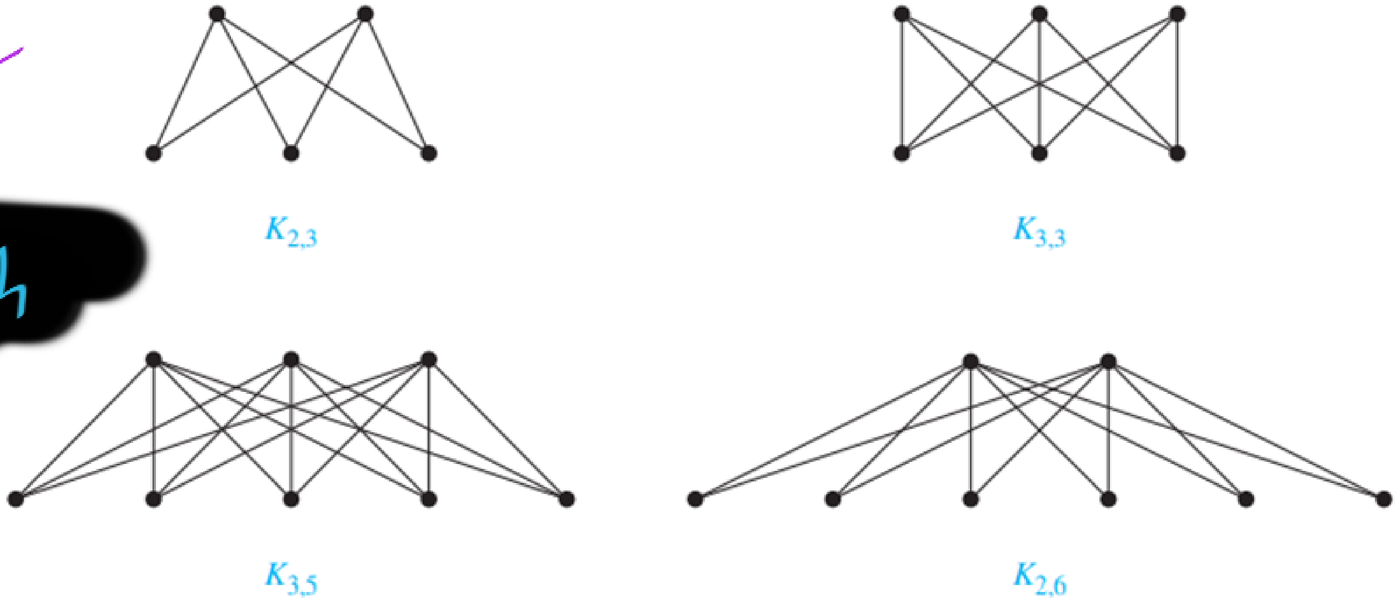
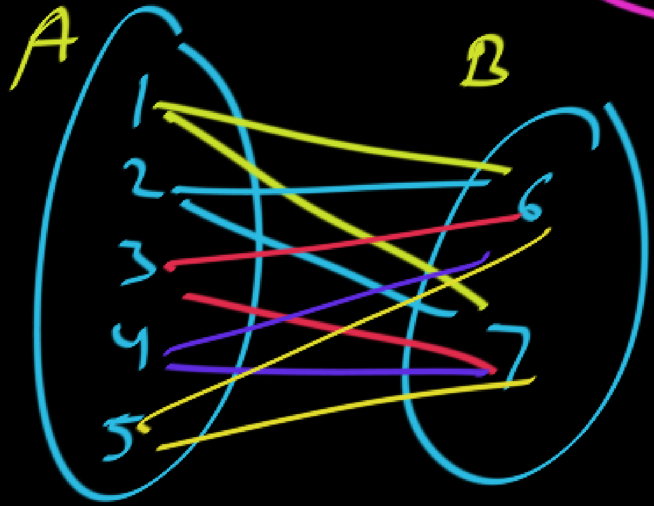


FIGURE 9 **Some Complete Bipartite Graphs.**

Q: Bipartite Graph, 7 vertices, 5 in part A, 2 in part B.

$G(\underbrace{A, B}_{V}, E)$

max number of Edges? $\Rightarrow$ $5 \times 2 = 10$



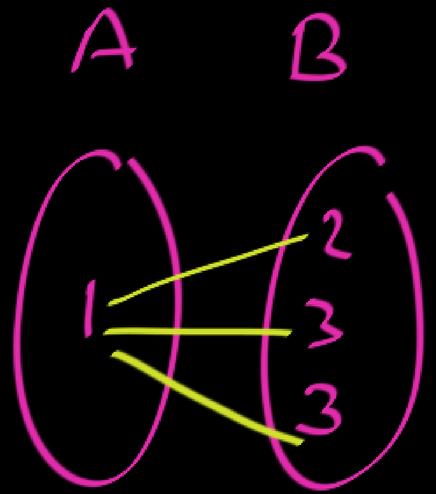
Create Complete Bipartite Graph.

$K_{5,2}$

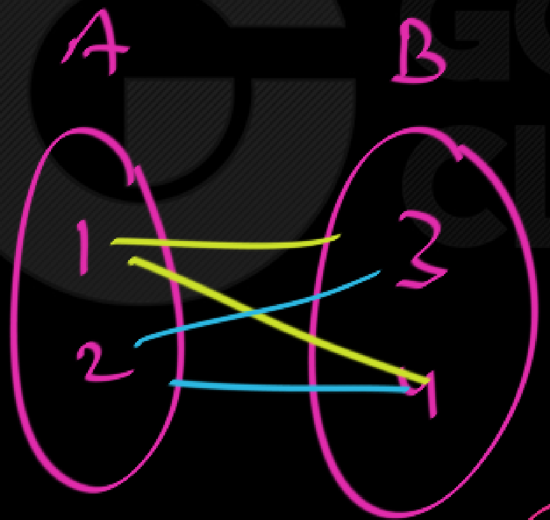
Maximum number of Edges in a Bipartite Graph on 4 vertices?

V → A, B

$|V| = 4$



max #Edges = 3

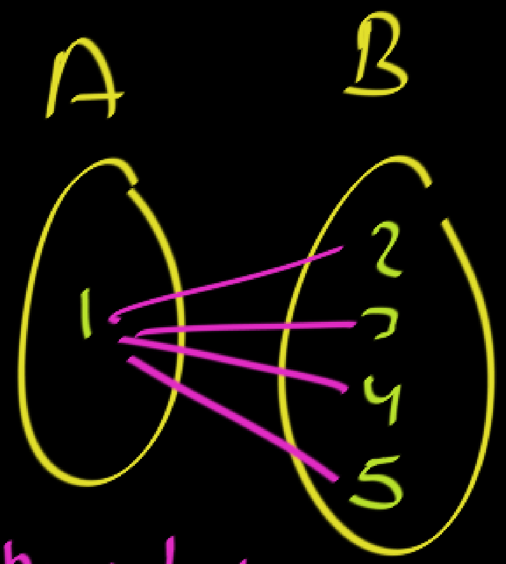


max #Edges = 4

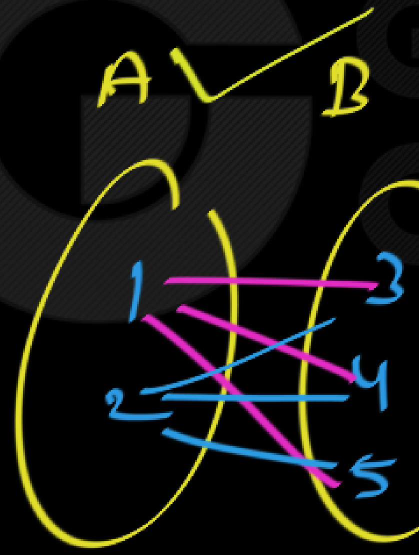
final Ans = 4

Maximum number of Edges in a Bipartite Graph on 5 vertices?

$G(\underbrace{A, B}_{V}, E) \Rightarrow |V| = 5$



max |E| = 4



max |E| = 6

max |E| = 0

final Answer = 6

Maximum number of Edges in a Bipartite Graph on n vertices?

$G(\underbrace{A, B}_{V}, E)$

$|V| = n$

A B

max $|E|$

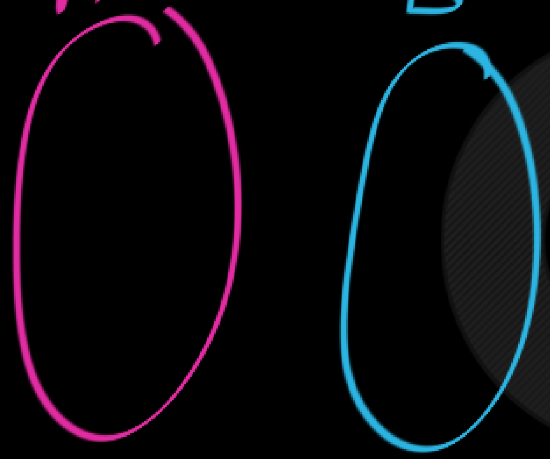
$= \underbrace{|A| \times |B|}$

maximize $|A| \times |B| = \underbrace{(m)(n-m)}_{\text{want to maximize}}$

m → variable

n → fixed

A B



$|A| = m$ then $|B| = n-m$

In Calculus: maxima, minima

$f = m(n-m)$ — "<u>To maximize</u>"

$$\frac{df}{dm} = \frac{d\, m(n-m)}{dm} = \frac{d\, mn - m^2}{dm} = \underbrace{n - 2m}$$

$$\boxed{n - 2m = 0} \Rightarrow \boxed{m = \frac{n}{2}}$$

A B

$n/2$ $n/2$

If n = odd = 7

$\frac{n}{2}$ $\frac{n}{2}$

3 4

$\left\lfloor \frac{n}{2} \right\rfloor$ $\left\lceil \frac{n}{2} \right\rceil$

Conclusion: Bipartite Graph $G(\underbrace{A, B}_{V}, E)$

$|V| = n$ ✓

To get maximum $|E|$ $\Rightarrow$

$\lfloor \frac{n}{2} \rfloor$ $\lceil \frac{n}{2} \rceil$

$\underbrace{\lfloor \frac{n}{2} \rfloor \times \lceil \frac{n}{2} \rceil}$ ✓

even

n vertices bipartite Graph can have maximum $\left(\frac{n}{2}\right)^2$ edges.

If a Graph on n vertices has $>\left(\frac{n}{2}\right)^2$ edges then it can not be Bipartite.

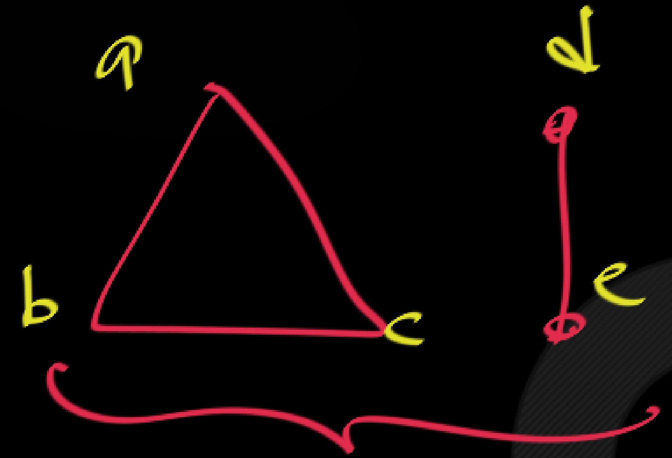
EXERCISES

- Prove that an n-vertex graph with more than $n^2/4$ edges is not bipartite.
- Find all n-vertex bipartite graphs with $n^2/4$ edges.
- Prove that for a non-empty regular bipartite graph the number of vertices in both parts is the same.

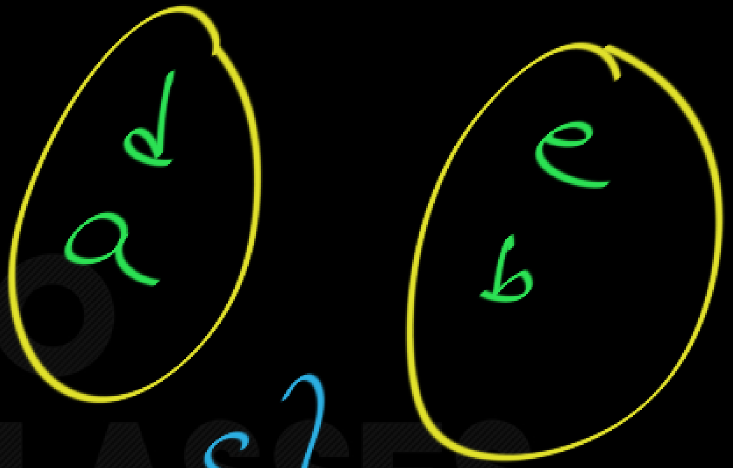
. G is bipartite if and only if each of its components is bipartite.

iff

No odd cycle



$\Longrightarrow$

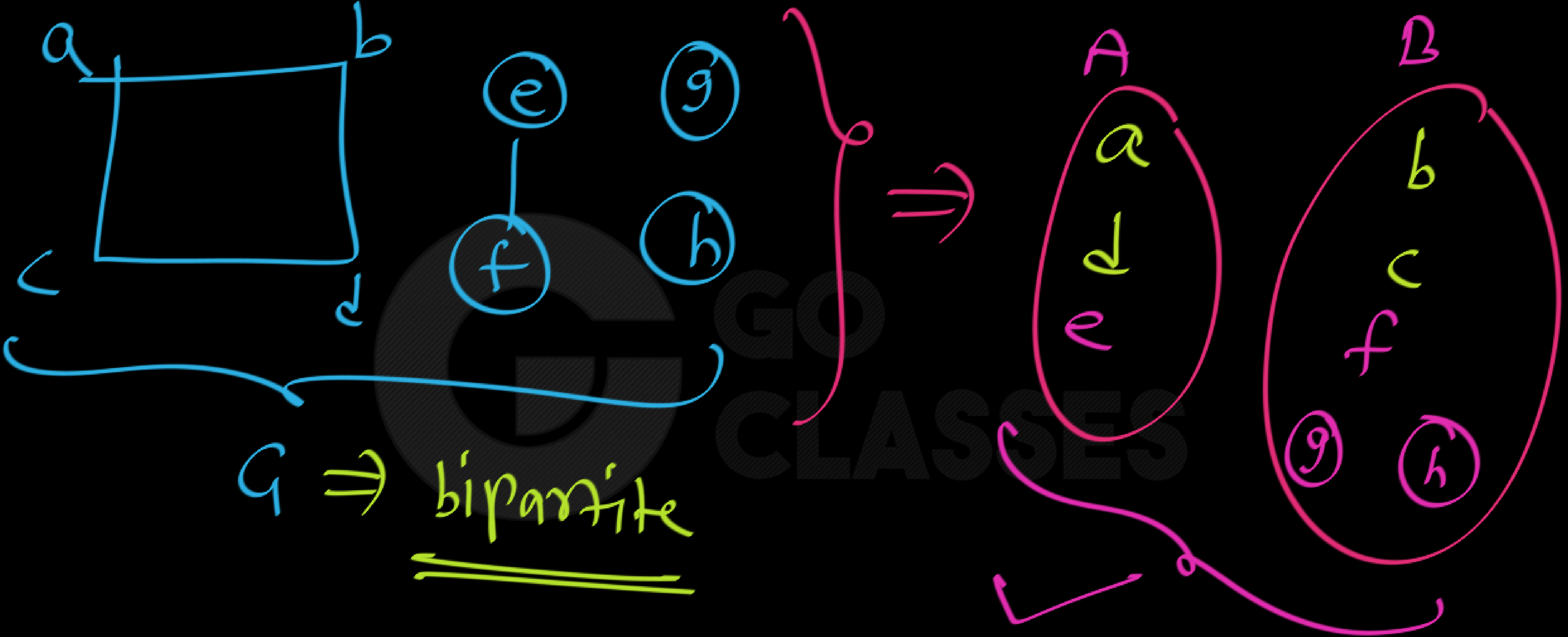


G

NOT bipartite

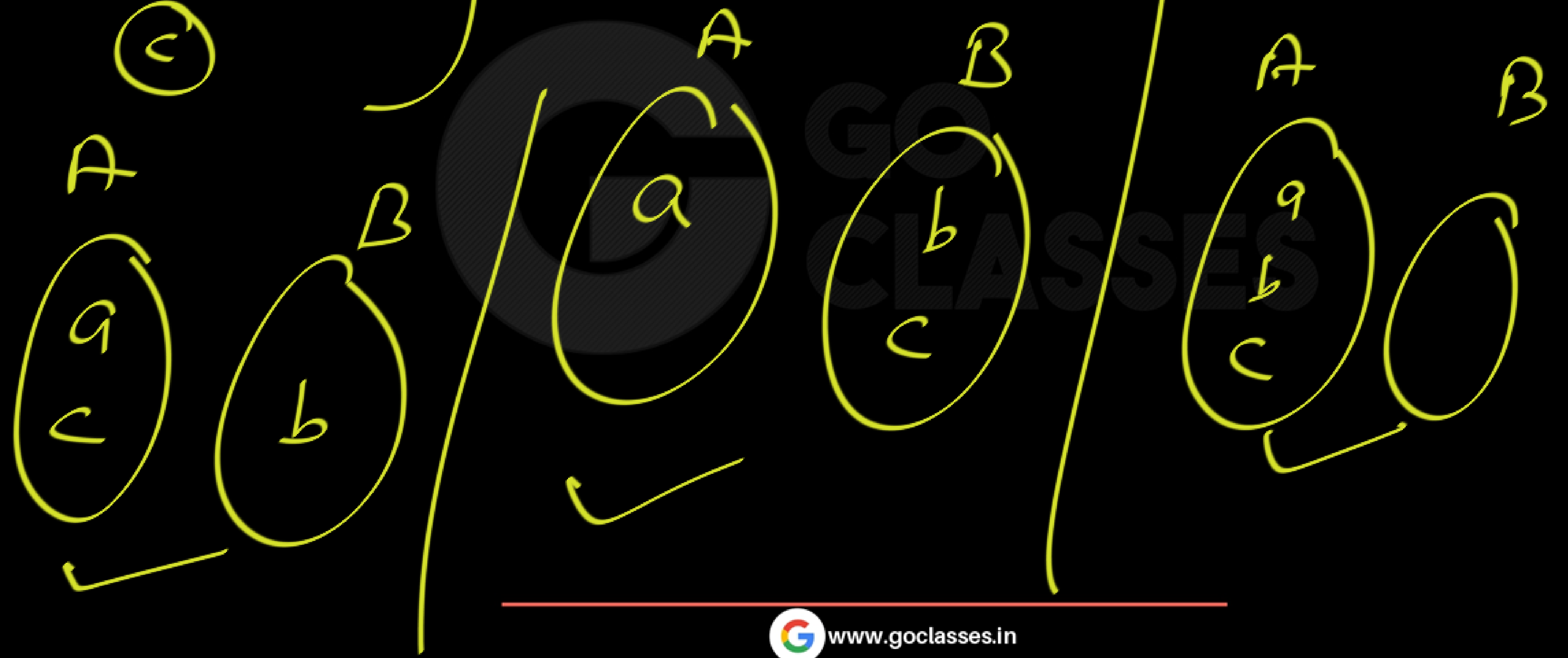
s?

NoT bipartite



$G \Rightarrow$ bipartite

a b c } G → bipartite



Which is Bipartite?

1. K_n
2. P_n
3. C_n
4. W_n
5. E_n
6. Q_n
7. $K_{m,n}$

Which is Bipartite? $\Rightarrow$ iff No odd length cycles

1. ~~K_n~~ (only for $n=1,2$)
2. P_n ~
3. ~~C_n~~ (only if n = even)
4. ~~W_n~~
5. E_n
6. Q_n
7. $K_{m,n}$

W_n :

"Parity" of a bit-string : (# of 1's in it.)

odd or even

P(0 0 1 0 1 0 1) = odd

P(0 0 0 1 0 1 0) = even

P(0 0 0) = even

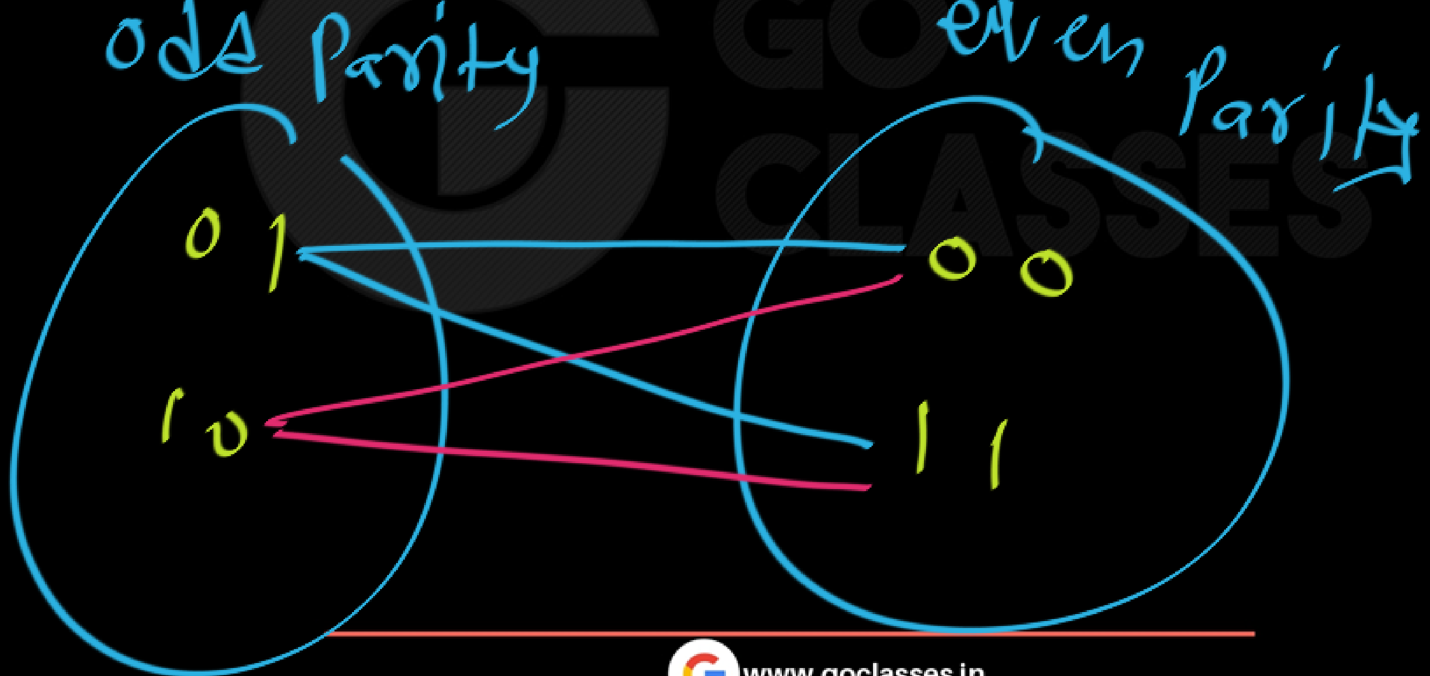
P(1 1 1 1 1) = odd

P(0) = even

P(1) = odd

$Q_n \Rightarrow$ Hypercube Graph on 2^n vertices

$\hookrightarrow$ Bipartite

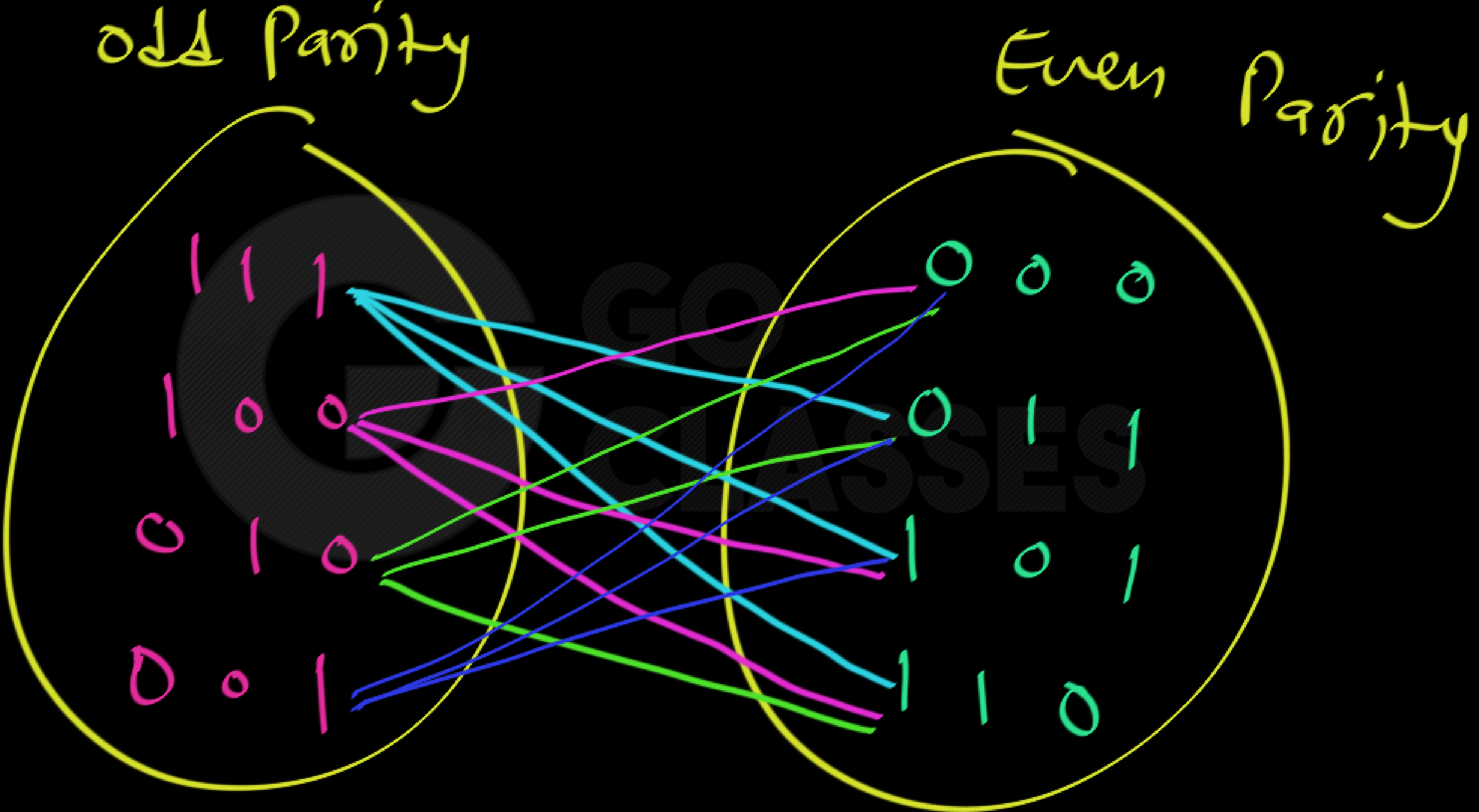


$\underbrace{0\,0\,1\,0\,1\,0}$ will be Adjacent to only odd parity string.

Even Parity String —/— Even Parity String

NOT Adjacent

Q3:



(c) Hypercubes are bipartite.

Explain why Q_n is bipartite in general.
[Hint: consider the parity of the number of 0's in the label of a vertex.]

Solution: Any two vertices with an even number of 0's differ in at least two bits, and so are non-adjacent. Similarly, any two vertices with an odd number of 0's differ in at least two bits, and so are non-adjacent. So let $V_1 = \{$ vertices with an even number of 0's $\}$ and $V_2 = \{$ vertices with an odd number of 0's $\}$.

Q_n, $\forall n$ is Bipartite :

$Q_n = G$ (odd parity strings, Even parity strings , E)

Part A

V

Part B

Q_n = Comp. bipartite Graph?, No.

GATE CSE 2014 Set 2

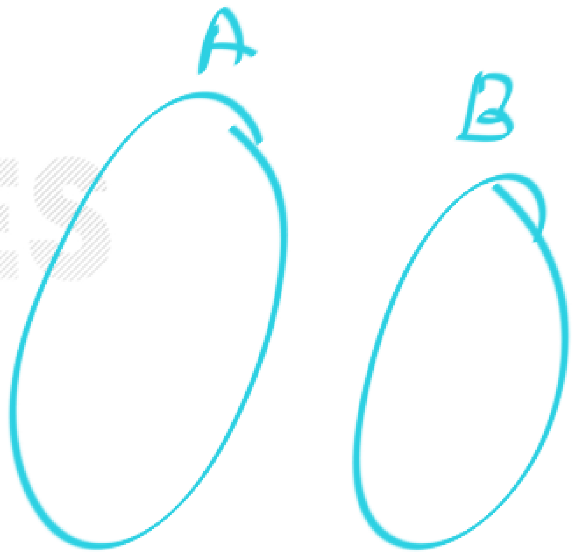
The maximum number of edges in a bipartite graph on 12 vertices is____

GATE CSE 2014 Set 2

The maximum number of edges in a bipartite graph on 12 vertices is___ 36

n

Max |E| in Bipartite Graph on n vertices $= \left\lfloor \frac{n}{2} \right\rfloor \times \left\lceil \frac{n}{2} \right\rceil$

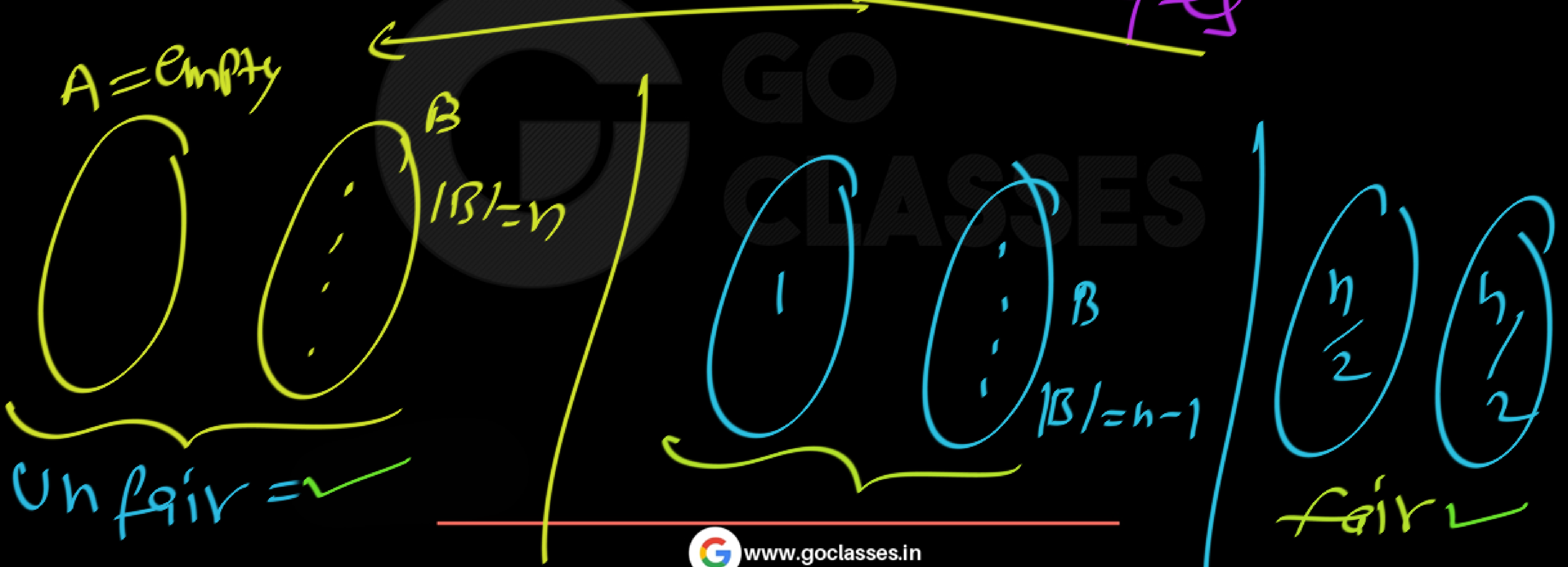


$|A| = 6$ $|B| = 6$

mostly, in Graph Theory or some other subjects;

"maximum, minimum" are generally found Either at Extreme "Unfair" Distribution OR Extreme "fair" Distribution

Bipartite Graph → maximum number of Edges



8 vertices bipartite Graph ; ⇒

Q : max $|E|$? ⟶ 4^2

To check :⇒ Corner Cases :

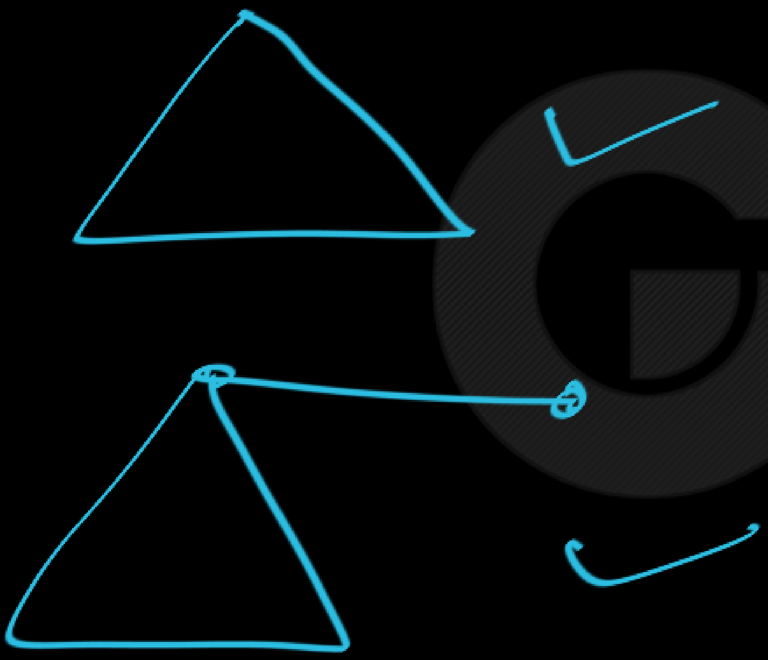
Graph Theory :

Next Topic :

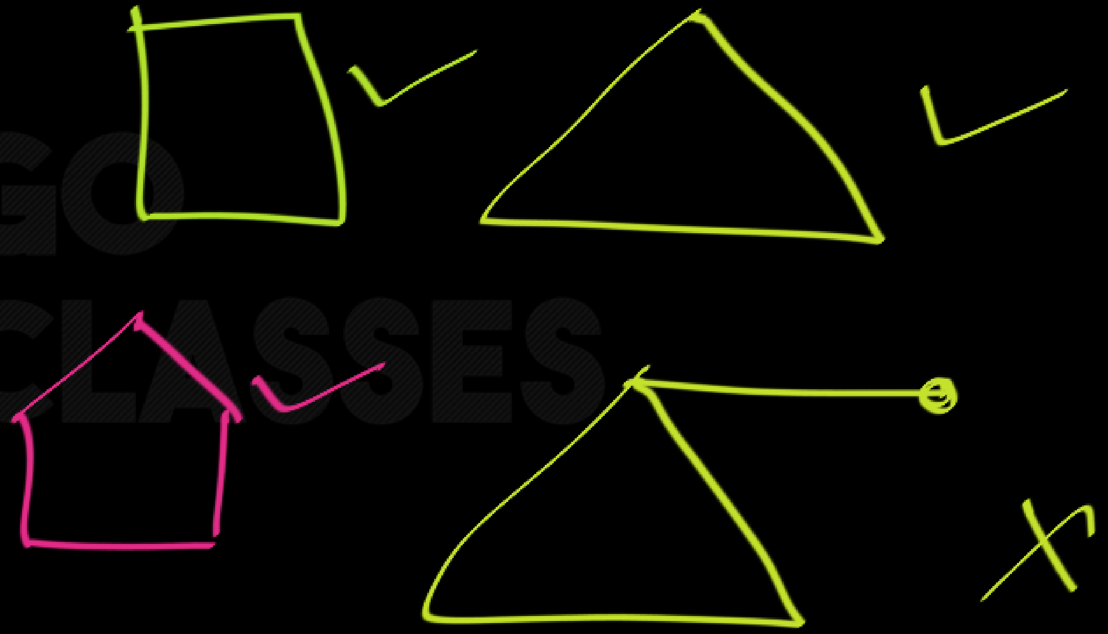
Cyclic, Acyclic Graphs

Website : https://www.goclasses.in/

Cyclic Graph

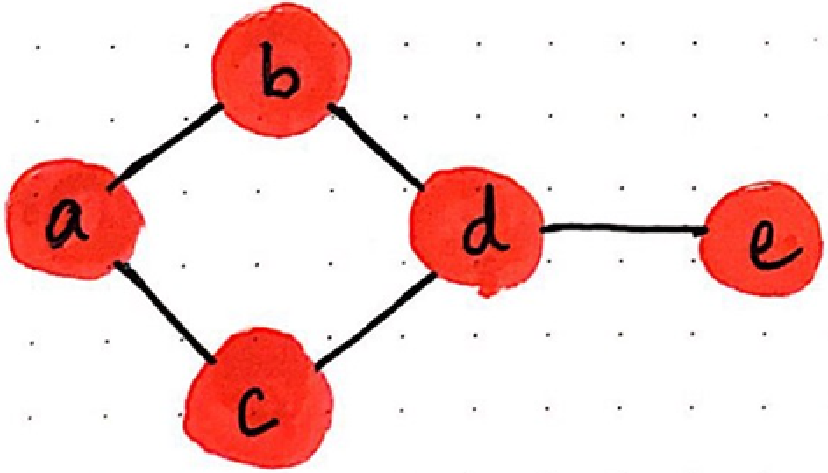


Cycle Graph

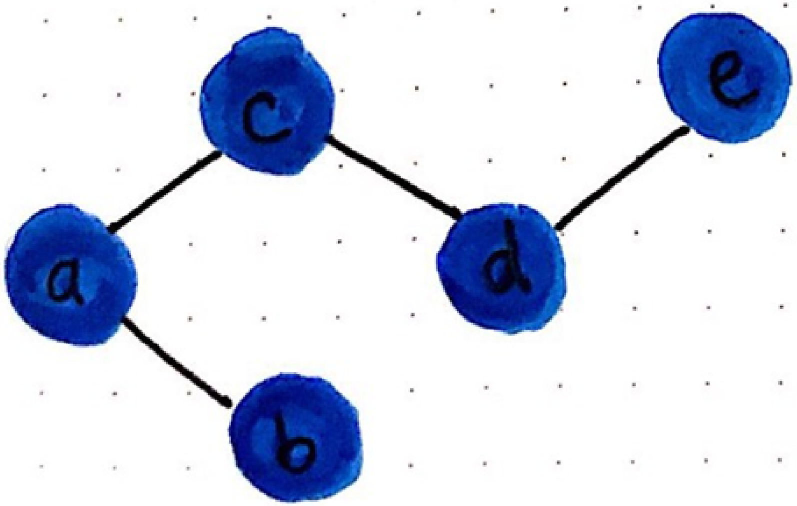


Cyclic Graph:

Graph Containing at least one Cycle.



A graph with at least one cycle is Known as a cyclic graph.



A graph with no cycles in it is Known as an acyclic graph.

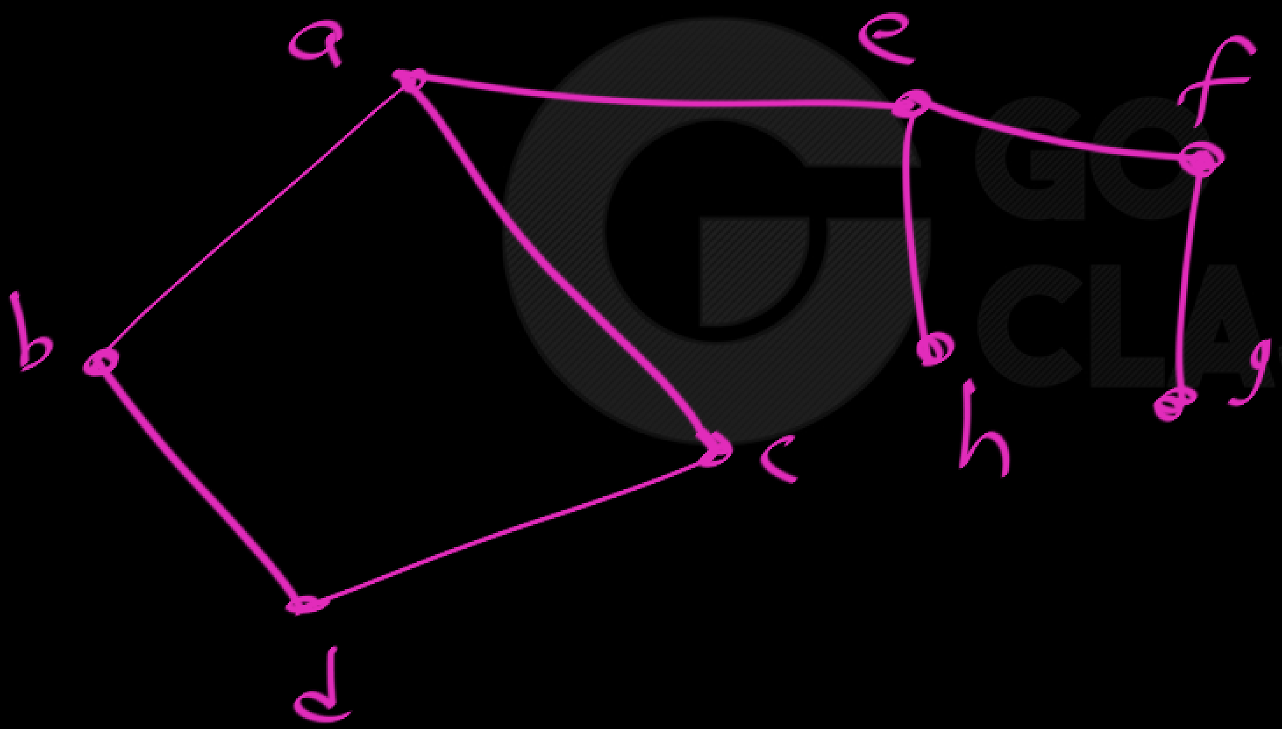
A cycle, in the context of a graph, occurs when some number of vertices are connected to one another in a closed chain of edges. A graph that contains at least one cycle is known as a cyclic graph. Conversely, a graph that contains NO cycles is known as an acyclic graph.

Cyclic Graph:

Graph Containing at least one Cycle.

i.e. there is some vertex where we can start off, follow a trail, and come back to the original vertex without repeating edges.

Cyclic Graph: → Contains at least one cycle.

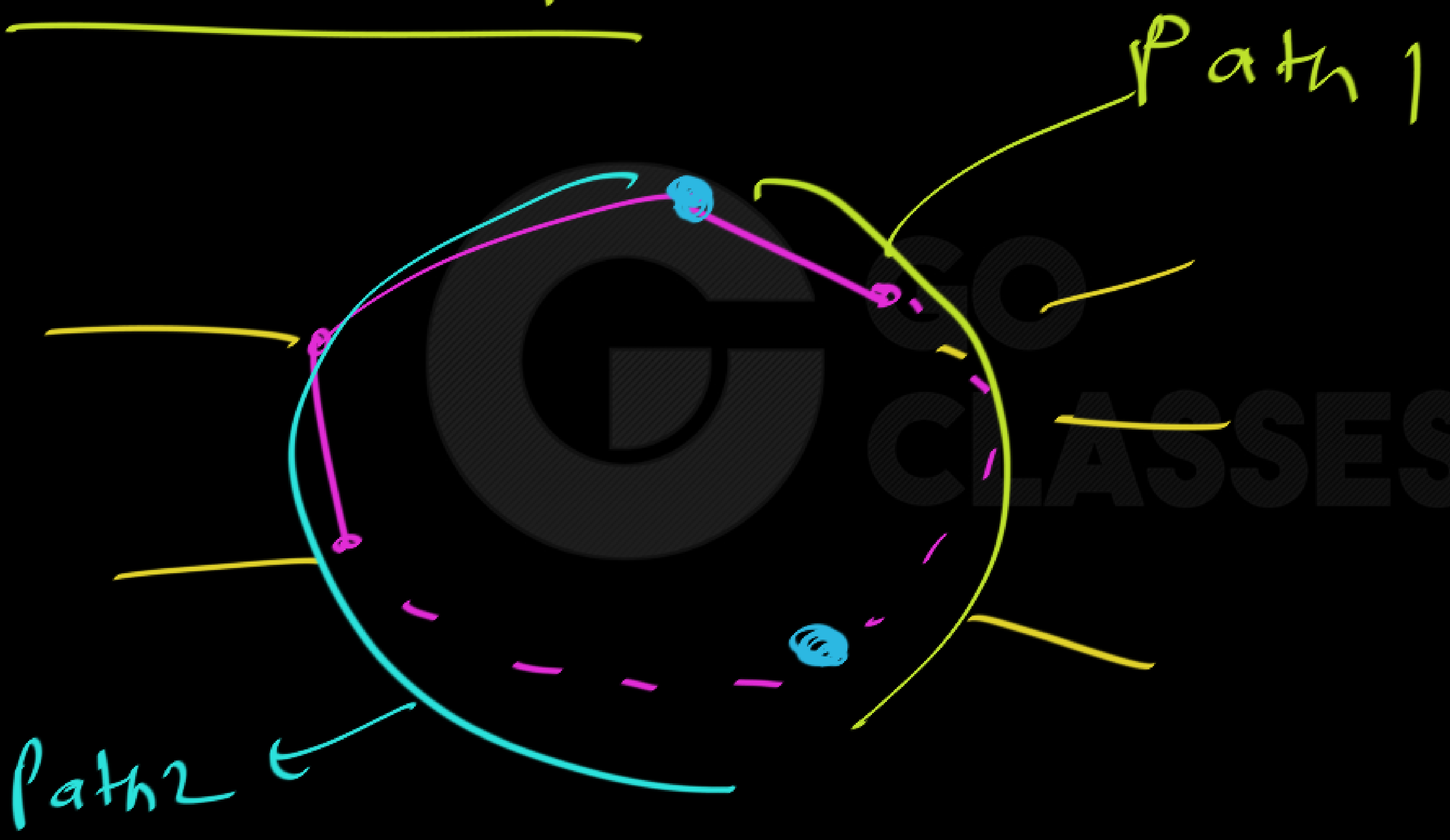


a-c Path?

a — c ✓

a — b — d — c ✓

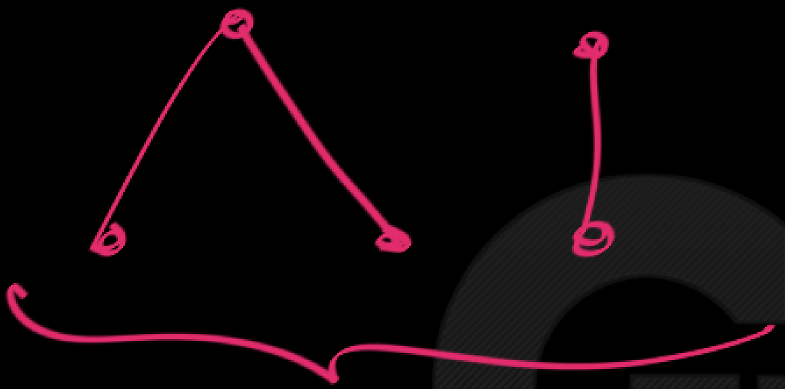
Cyclic Graph:



Cyclic Graph:

Graph Containing at least one Cycle.

i.e. there are at least two vertices which have more than one path between them.



Every two vertices have At most one Path.

G

Acyclic

Cyclic Graph:

Definition 1: Graph Containing at least one Cycle.

Definition 2: There are at least two vertices which have more than one path between them.

Definition 3: There is some vertex where we can start off, follow a non-empty trail, and come back to the original vertex without repeating edges.

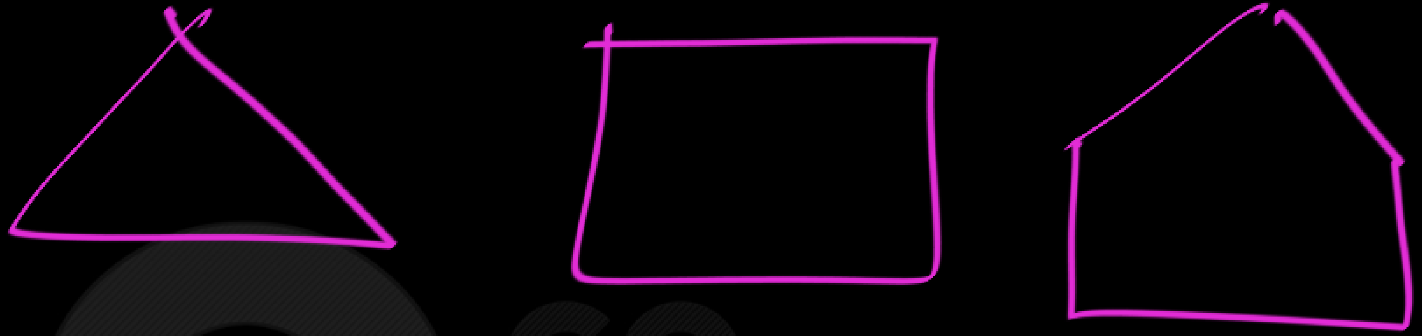
Acyclic Graph:

Definition 1: Graph Containing NO Cycle.

Definition 2: Every pair of vertices have at most one path between them.

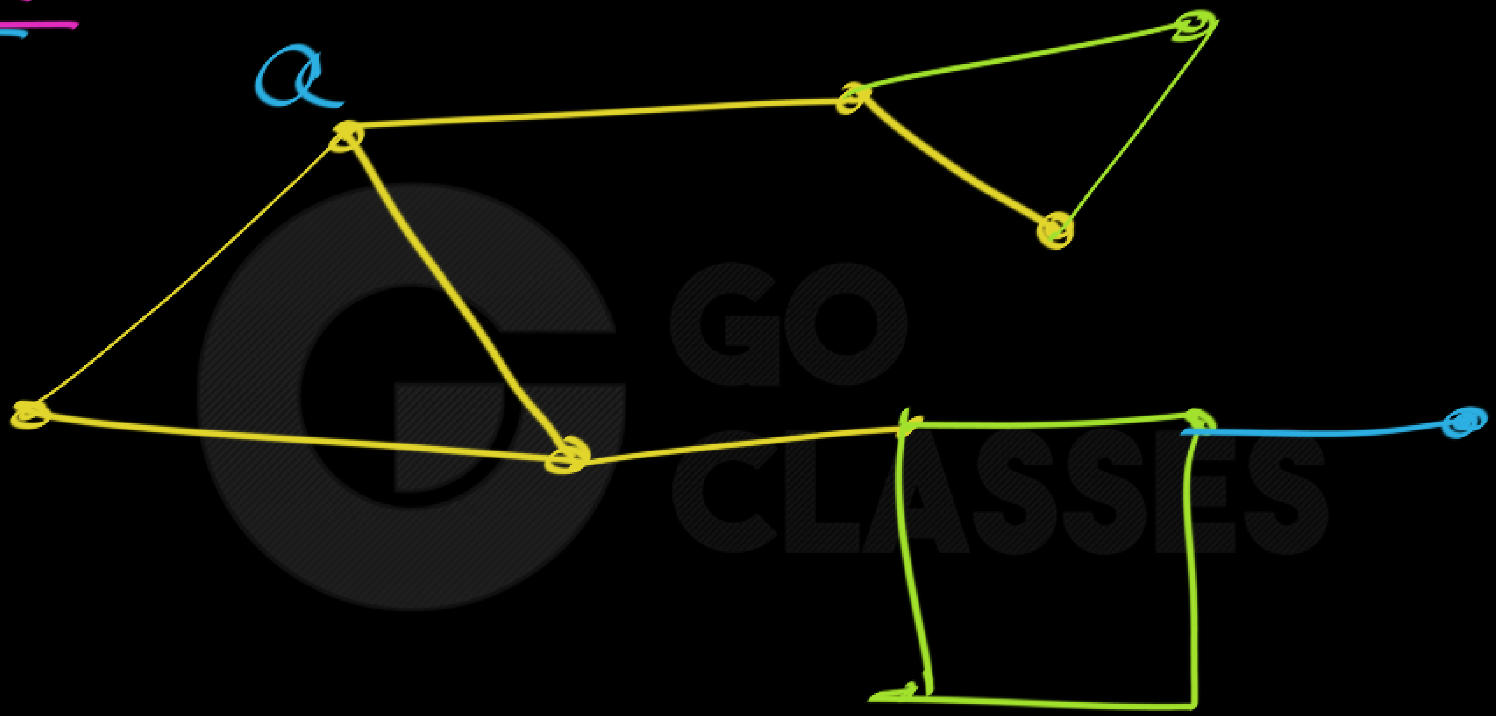
Definition 3: There is NO vertex where we can start off, follow a non-empty trail, and come back to the original vertex without repeating edges.

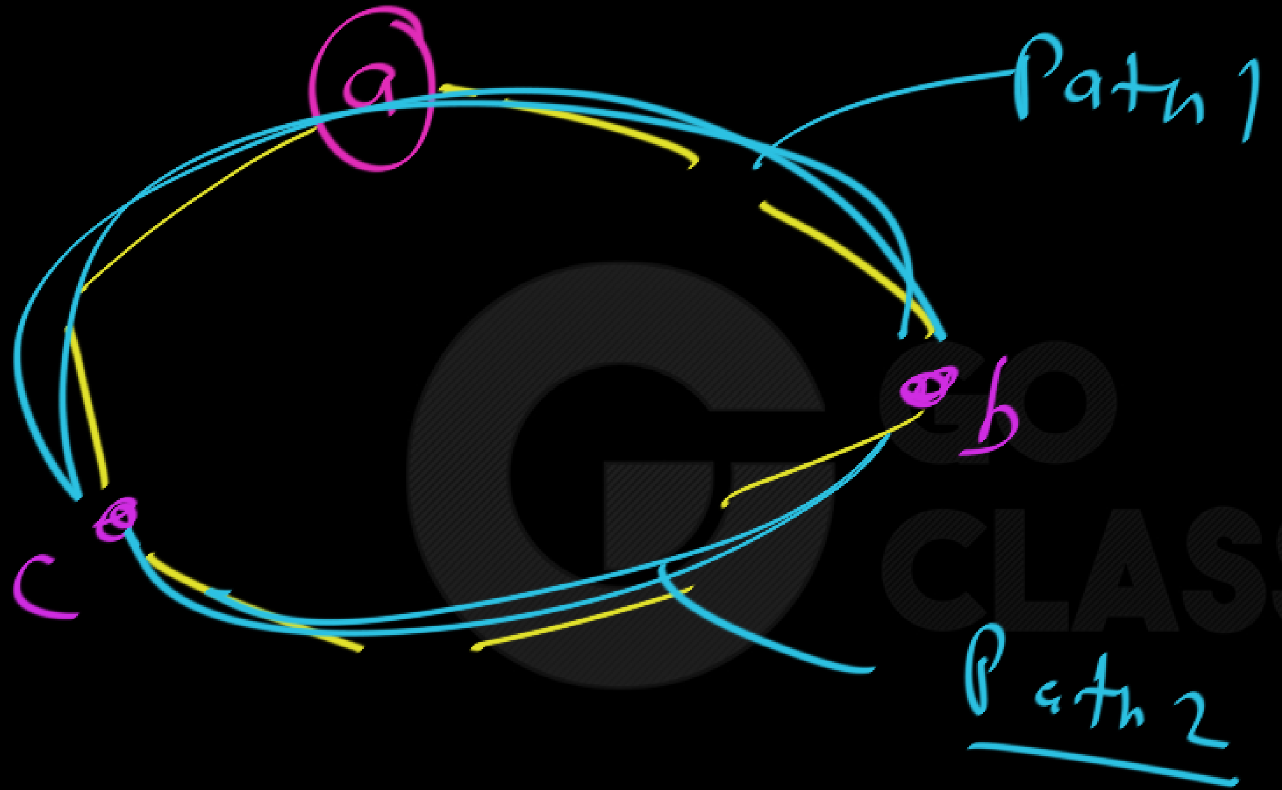
Cycle:



at least "three vertices" in a cycle.

Cyclic :




a
b
c
Path 1
Path 2

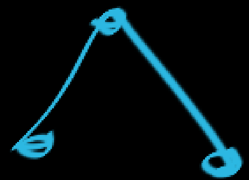
Graph Theory :

Next Topic :

Tree

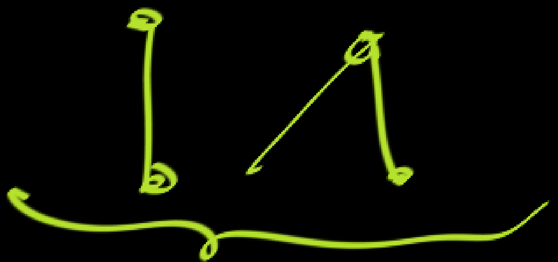
Website : https://www.goclasses.in/

Tree — "Connected, Acyclic" Graph



forest : Acyclic Graph

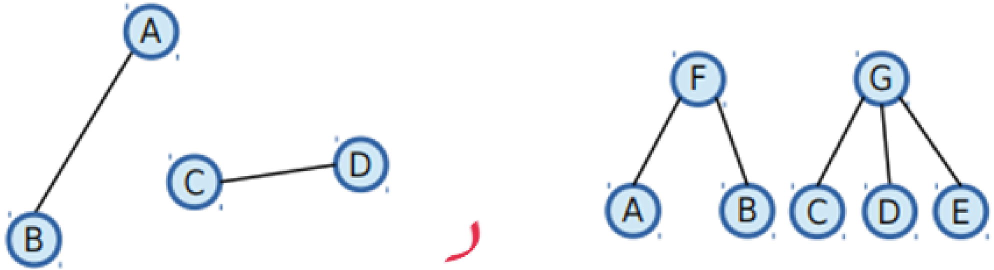
Collection of Trees

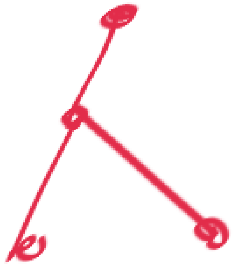


Graphs without cycles

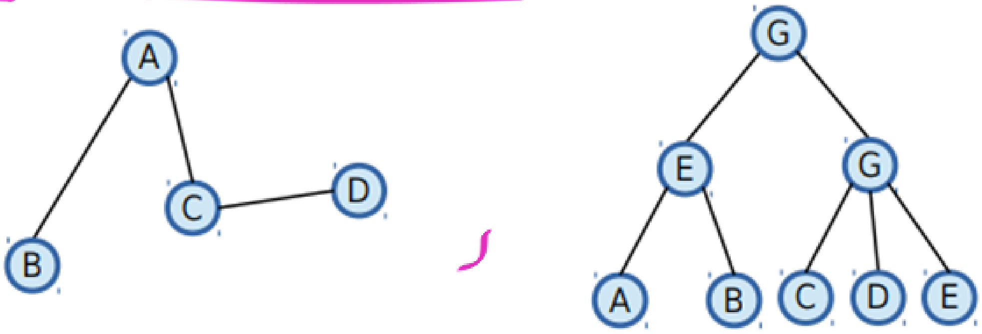
- A forest is an undirected graph with no cycles

Acyclic





- A tree is a connected forest (← definition)



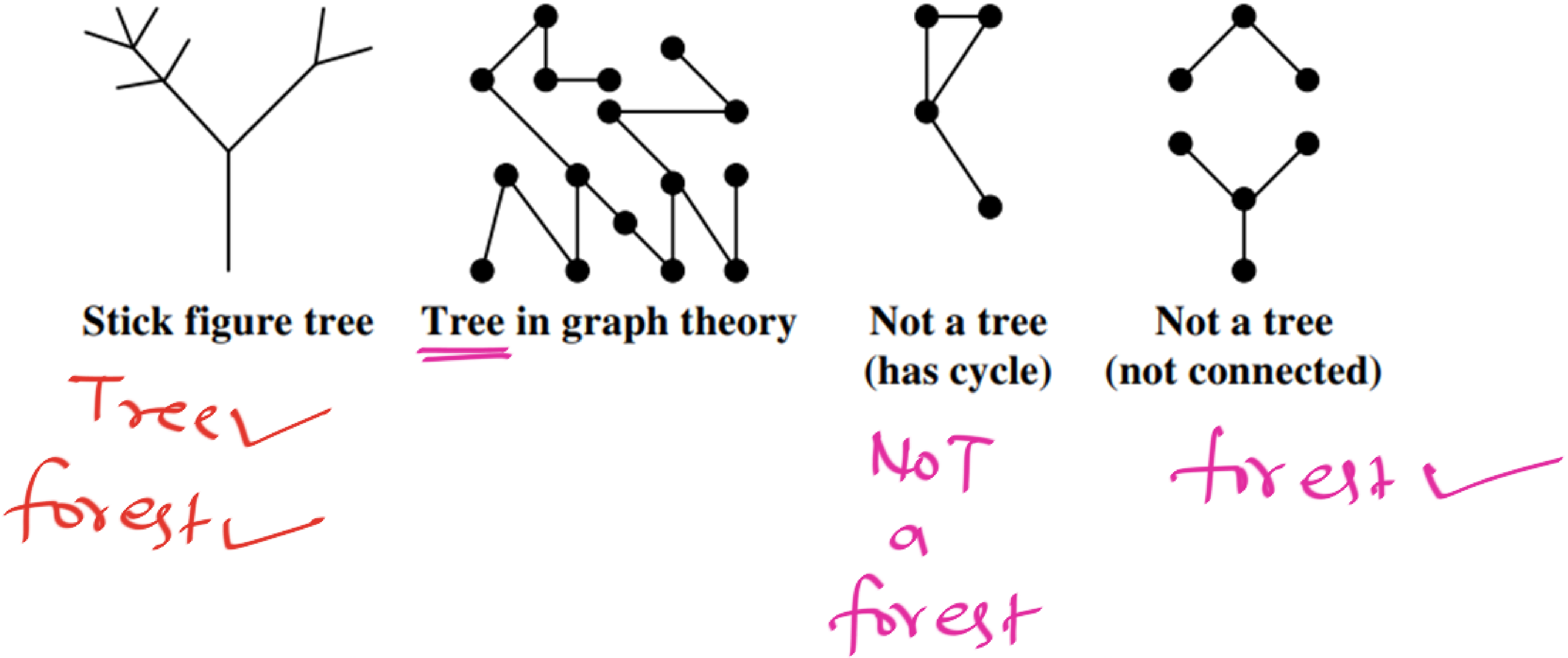
Trees as Graphs

When talking about graphs,
we say a tree is a graph that is:

- Undirected
- Acyclic
- Connected

So all trees are graphs, but not all graphs are trees

Definition 28. A connected graph without cycles is called a *tree*.



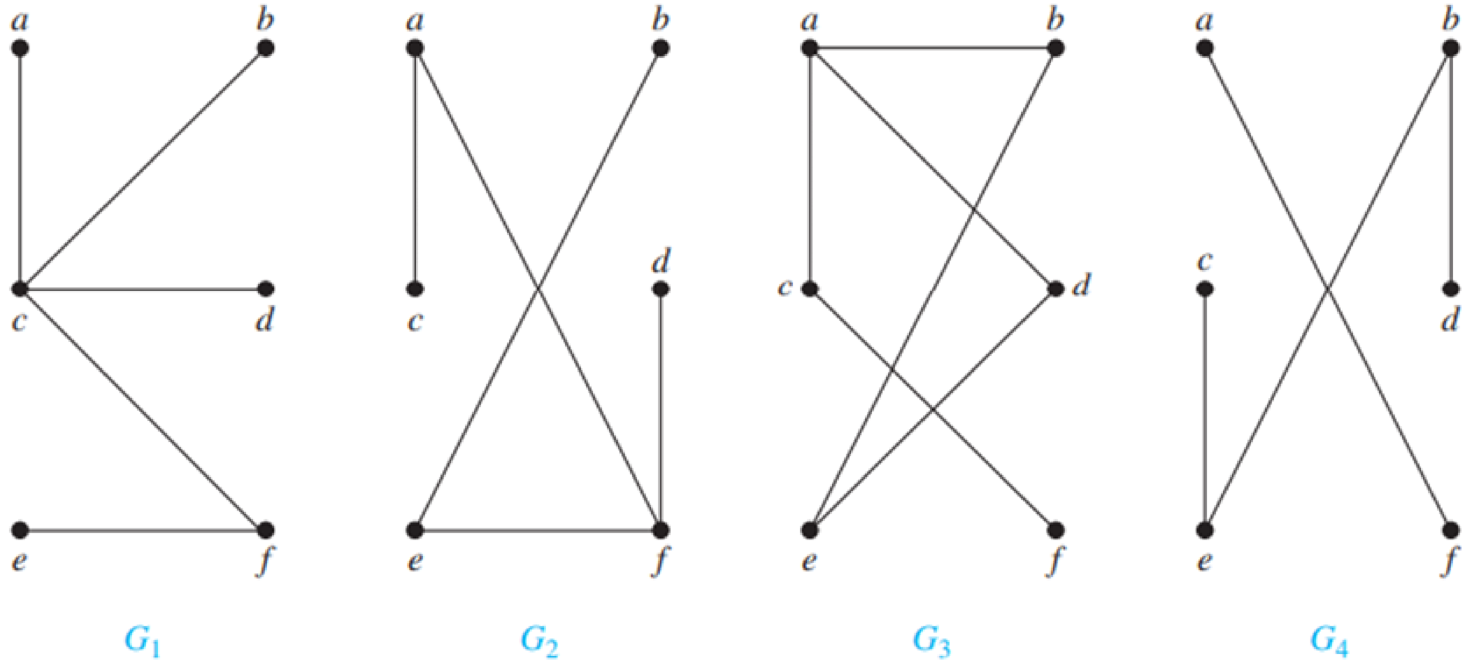
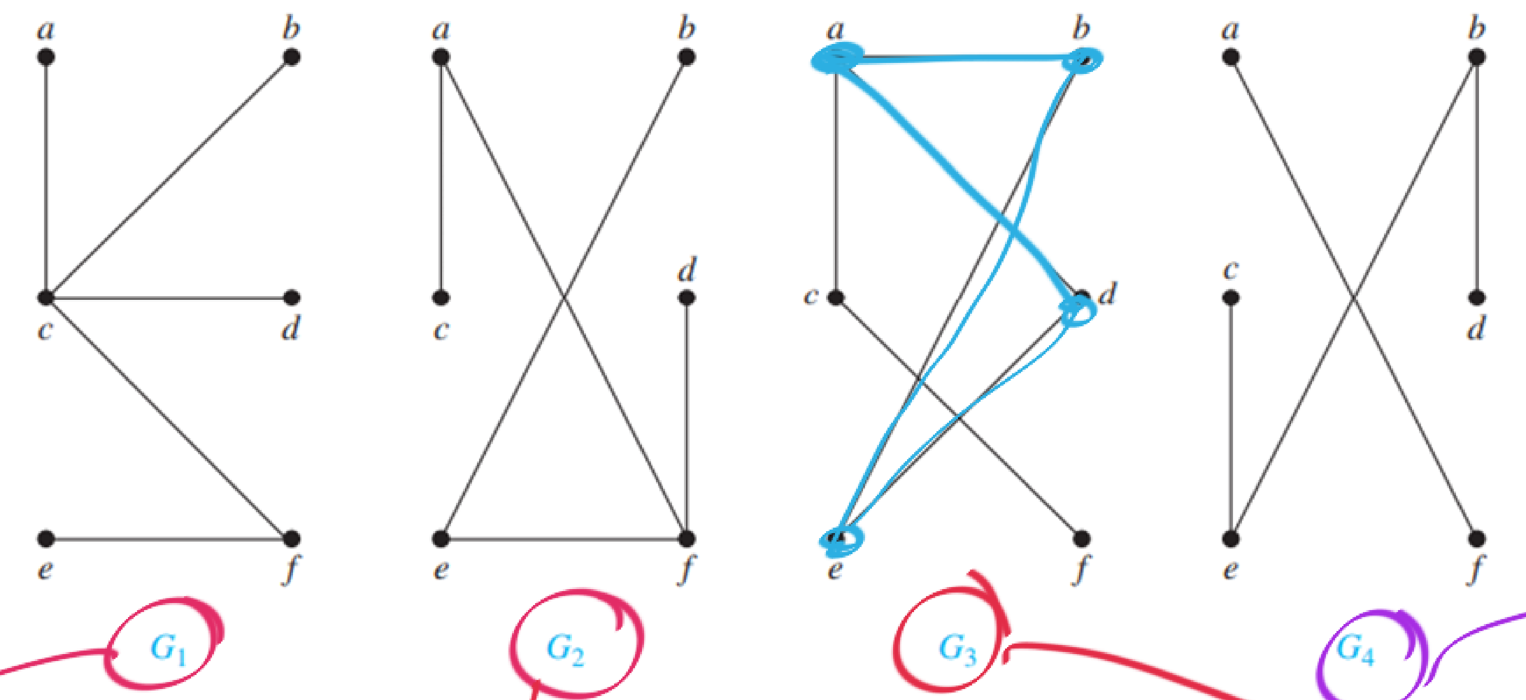


FIGURE 2 **Examples of Trees and Graphs That Are Not Trees.**

abcd cycle



Tree ✓

Forest ✓

Tree, forest

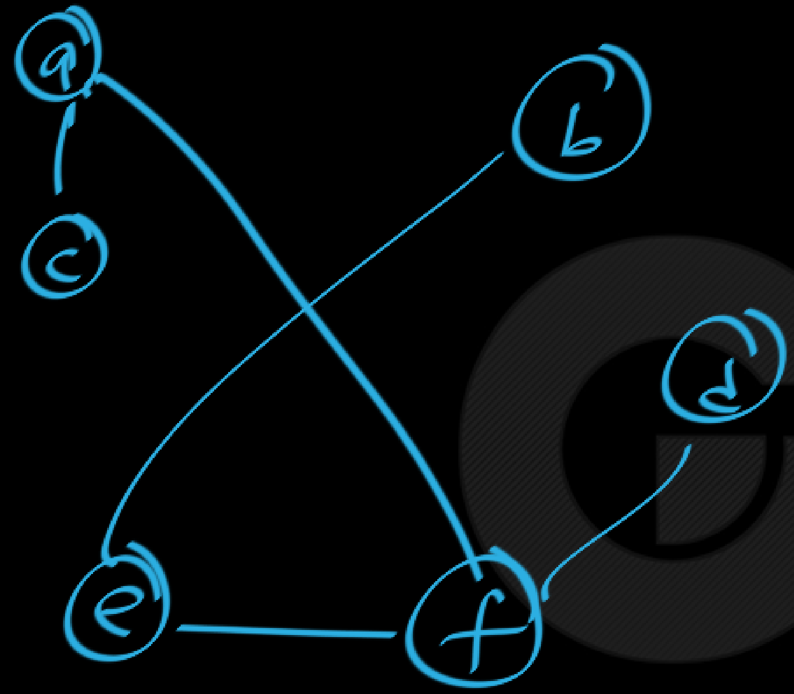
forest ✓

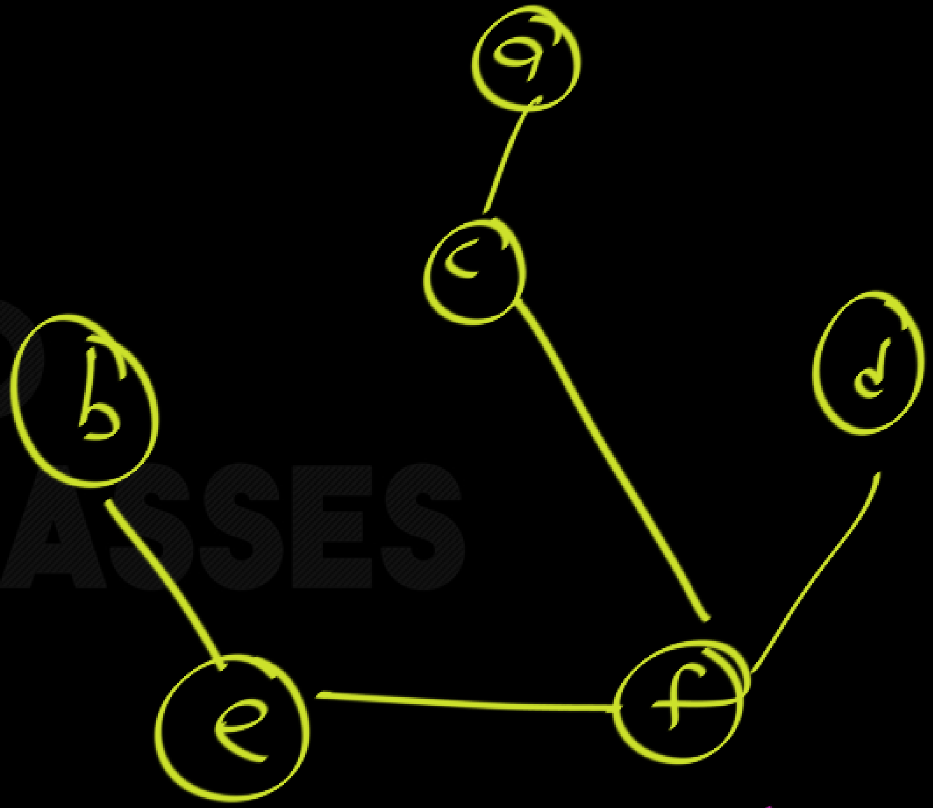
Tree ✗

Not Tree

Not forest

FIGURE 2 Examples of Trees and Graphs That Are Not Trees.





Acyclic ; Connected


A TREE WITH A CYCLE

Definition 29. A graph every connected component of which is a tree is called a *forest*. In other words, a forest is a graph without cycles. Forests are also known as *acyclic graphs*.

Definition 32. In a tree a vertex of degree 1 is called a *leaf*.

Every Tree is a forest.

Tree iff $n = 1, 2$

- Which of the following graphs are forests:

- Which of the following graphs are trees: P_n, C_n, K_n:

Tree

NOT forest
NOT Tree

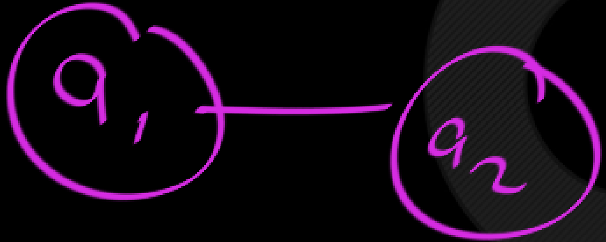
Not forest
NOT Tree

G_1

Tree ✓

Connected ✓, Acyclic ✓

G_2



Tree ✓

This is one graph with three connected components.

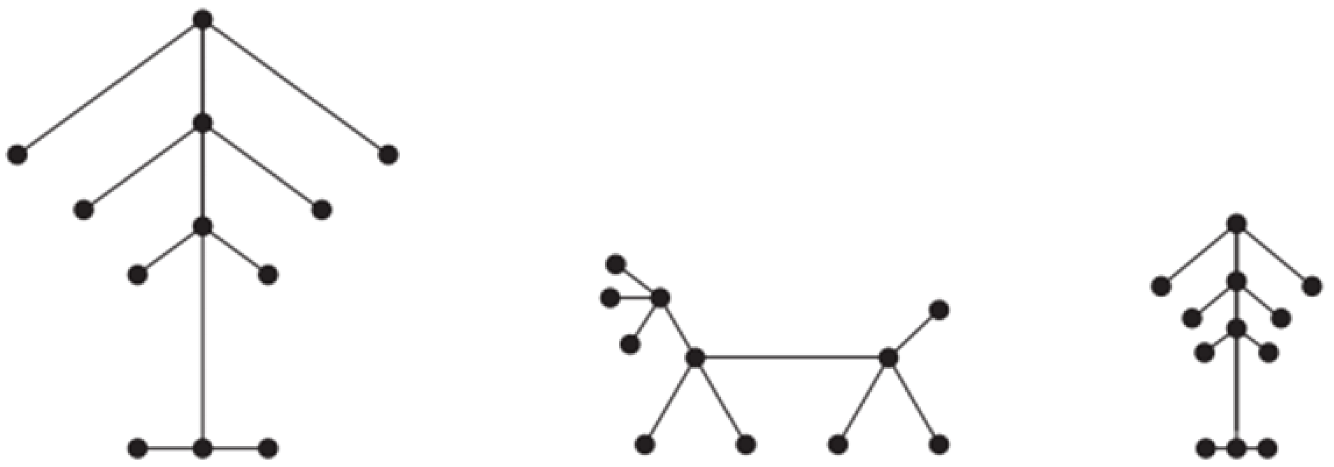


FIGURE 3 **Example of a Forest.**

Complete Analysis of Tree:

forest

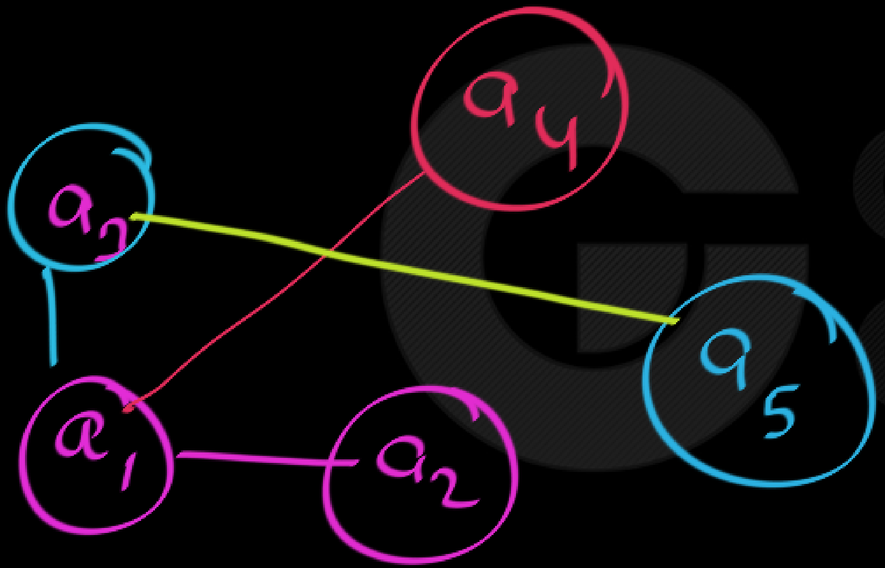
① Tree = Connected, Acyclic

Undirected

② |E| in Tree?

$$\boxed{|E| = |V| - 1}$$

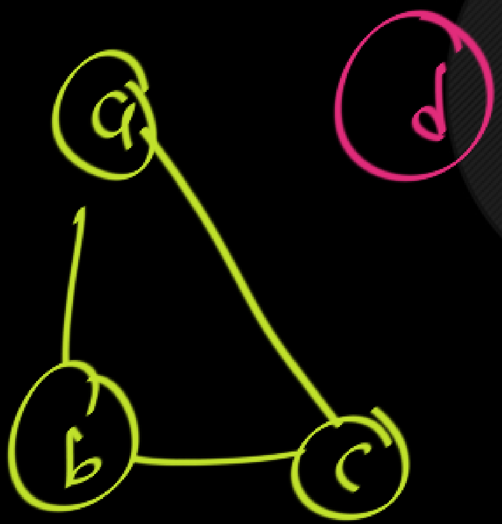
Tree on n vertices :



$\lvert V \rvert = n$	$\lvert E \rvert$
1	0
2	1
3	2
4	3
5	4
n	$n-1$

Q: $\underbrace{\text{Graph } n \text{ vertices, } n-1 \text{ Edges}}$ → Tree ?

Q: Graph n vertices, n-1 Edges → Tree? No



$n = 4$
$|E| = 3$

But NOT Tree.

Tree $\longrightarrow$ $\underbrace{|E| = |V| - 1}$

Q: Connected Graph on n vertices,

$\underbrace{\text{Connected Graph on } n \text{ vertices, } n-1 \text{ Edges}}$

↓

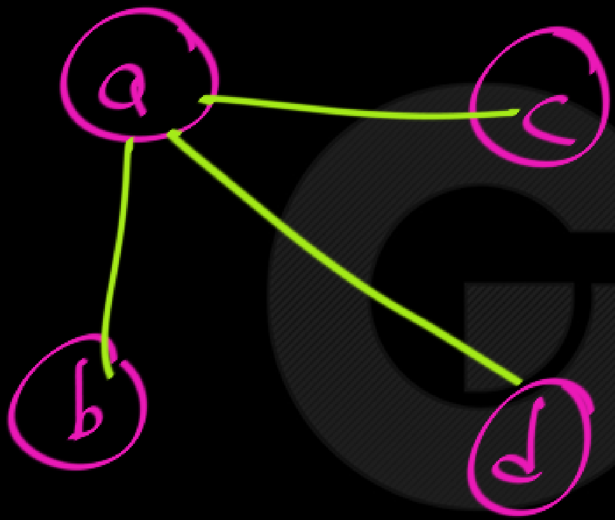
Tree?

Q: Connected Graph on n vertices,

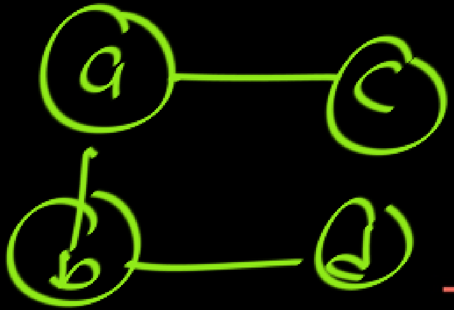
n-1 Edges

Tree? = Yes

$n=4$ = "Connected Graph" with 3 Edges.



Tree



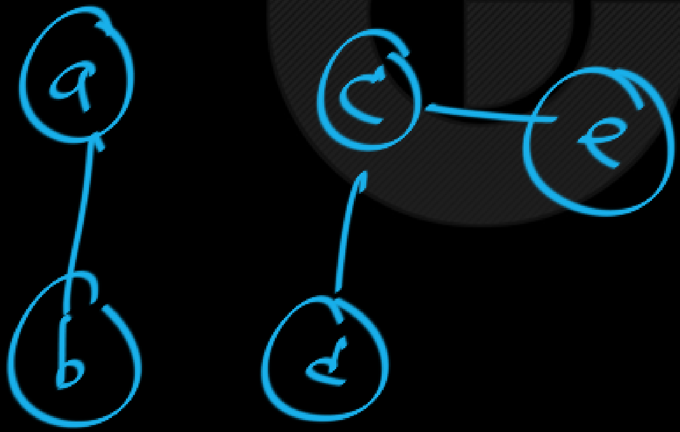
Tree

Tree $\equiv$ Connected Acyclic

$\equiv$ Connected with $n-1$ Edges

$\equiv$

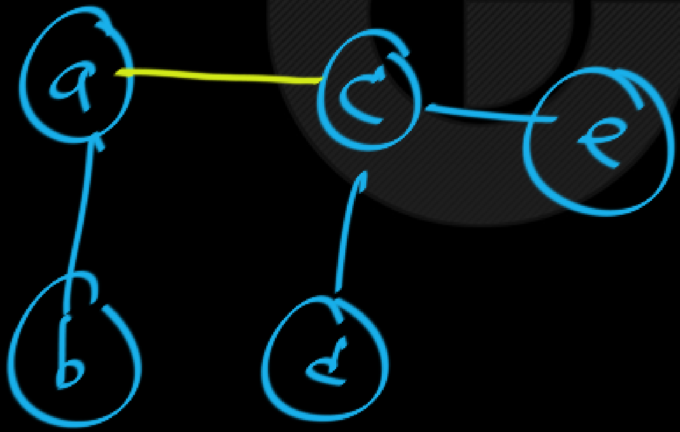
Q: Is this "maximally Acyclic" Graph on 5 vertices?



Acyclic ✓

maximally Acyclic on 5 vertices?

Q: Is this "maximally Acyclic" Graph on 5 vertices?



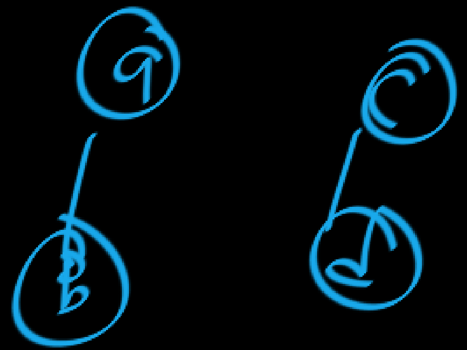
Acyclic ✓

maximally Acyclic on 5 vertices? No

Non maximally Acyclic

Maximally Acyclic Graph on n vertices = Tree forest ≡ Acyclic Graph

b/w any two vertices we have At most one path.



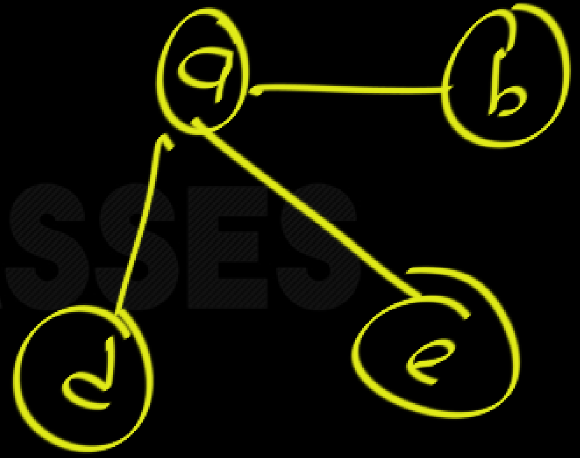
Tree :

b/w "any two vertices" we have Exactly one path.

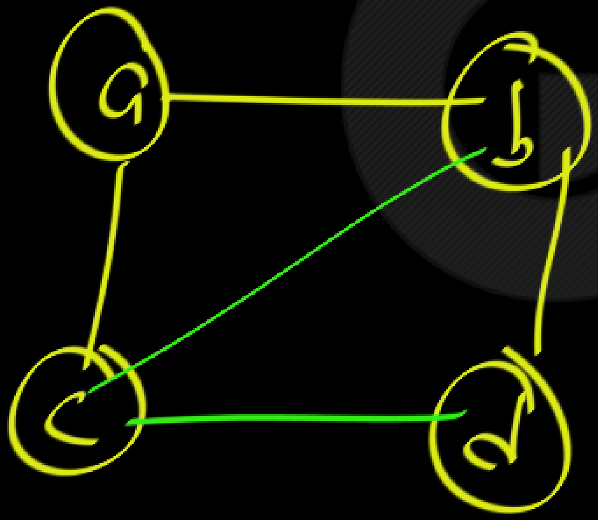
= Connected, Acyclic

Tree + one more Edge (Any Edge)

Cyclic Graph

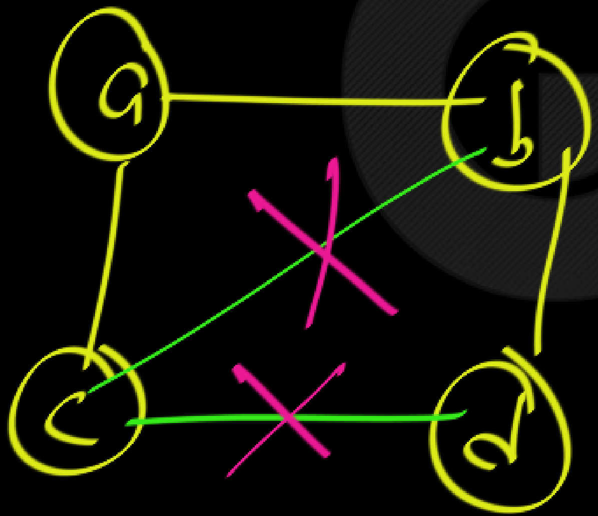


Q: Is this Graph minimally Connected on 4 vertices?



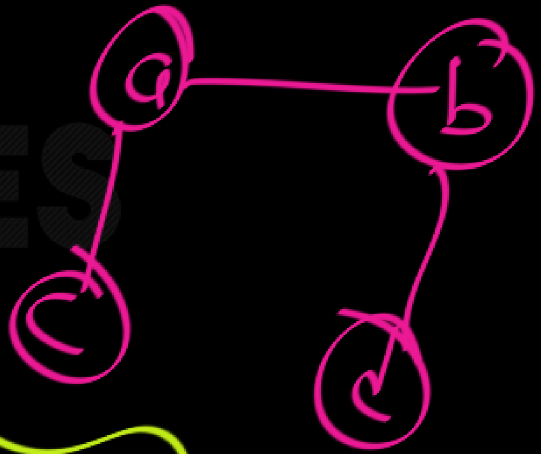
?

Q: Is this Graph minimally connected on 4 vertices?



?

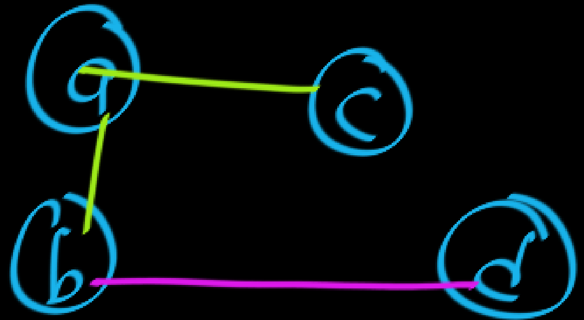
No



minimally connected

Tree ≡ Connected with minimum #Edges.

Tree ≡ Acyclic with maximum #Edges



Theorem 30. *The following statements are equivalent for a graph T:*

(1) *T is a tree.*

(2) *Any two vertices in T are connected by a unique path.*

(3) *T is minimally connected, i.e. T is connected but $T - e$ is disconnected for any edge e of T.*

(4) *T is maximally acyclic, i.e. T is acyclic but $T + uv$ contains a cycle for any two non-adjacent vertices u, v of T.*

(5) *T is connected and $|E(T)| = |V(T)| - 1$.*

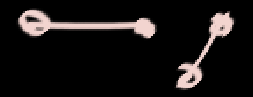
(6) *T is acyclic and $|E(T)| = |V(T)| - 1$.*

Q: which of the following is Tree

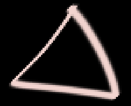
1. Acyclic Graph
2. Connected Graph
3. Acyclic Connected Graph
4. Connected Graph with minimum Edges Possible
5. minimally Connected

Q: Which of the following is Tree

1. Acyclic Graph



2. Connected Graph



3. Acyclic Connected Graph

4. Connected Graph with minimum Edges Possible

5. minimally Connected

Q: which is Tree?

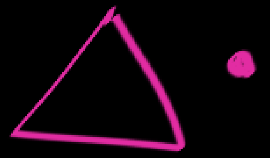
(a) n-vertices Graph with n-1 Edges

(b) " Connected " " " "

(c) maximally Acyclic Graph

(d) Every two vertices have unique Path b/w them.

Q: which is Tree ;



$n = 4$

$|E| = 3$

(a) n-vertices Graph with n-1 Edges

(b) " Connected " " " "

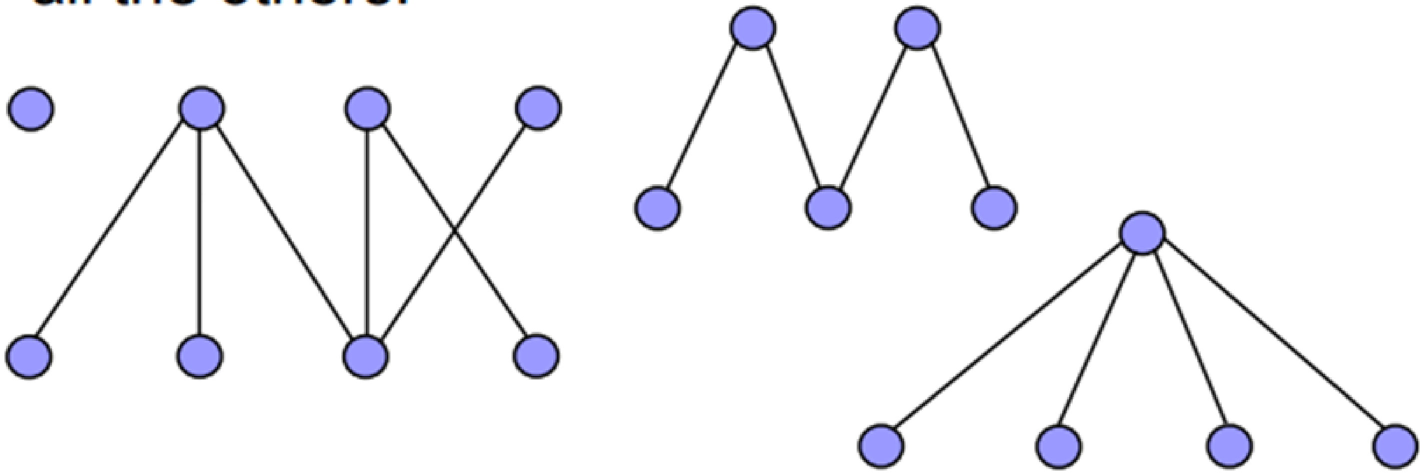
(c) Maximally Acyclic Graph

(d) Every two vertices have unique Path b/w them.

Properties

- Tree: a connected graph with no cycle (*acyclic*)
- Forest: a graph with no cycle
- Paths are trees.
- Star: A tree consisting of one vertex adjacent to all the others.

$K_{1,n}$



Degree Summation of Tree :

$$\sum_{\forall v \in V} \text{Deg}(v) = 2|E| = \underbrace{2(n-1)}$$

$$\text{Total Degree} = 2(n-1)$$

Q:

Every Tree T with at least two vertices has at least two vertices of Degree 1.

Proof by Contradiction: Tree, $n \geq 2$

"Assume" we Don't have at least two vertices of Degree 1.

$$\underbrace{\#\text{Deg}(1) \text{ vertices} \leq 1}$$

#vertices with Degree at least 2 $\geq n-1$

Tree $\longrightarrow$ Connected

So Every vertex Degree ≥ 1

$\underbrace{a_1 \quad a_2 \quad - - - \quad a_n}$

Total Deg $\geq \underbrace{(n-1)2+1}$ Contradiction

Q: Tree, $n \geq 2$

To prove: At least 2 vertices have Deg = 1

for Contradiction: "Assume"

$\#$ Deg(1) vertices ≤ 1 ✓

Tree $\rightarrow$ Connected $\boxed{n \geq 2}$ Given

So, $\text{Deg}(v) \geq 1 \quad \forall v \in \text{Tree}$

So, #Deg(1) vertices ≤ 1

All vertices have Deg ≥ 1

#Vertices with (Degree ≥ 2) $\geq n-1$

$$\boxed{\text{Total Degree} = 2(n-1)+1}$$

Contradiction

$a_1, \underbrace{a_2, a_3, - - -, a_n}_{Dep \geq 2}$

$Dep = 1$

Corollary 31. *Every tree T with at least 2 vertices has at least 2 vertices of degree 1.*

Proof. By Handshake Lemma, $\sum deg(v)_{v \in V(T)} = 2|E(T)|$ and by Theorem 30, $|E(T)| = |V(T)| - 1$. Therefore, $\sum deg(v)_{v \in V(T)} = 2|V(T)| - 2$. Therefore, T must contain at least 2 vertices of degree 1. $\square$

Proof Let the number of vertices in a given tree T be $n(n > 1)$. So the number of edges in T is $n-1$. Therefore the degree sum of the tree is $2(n-1)$. This degree sum is to be divided among the n vertices. Since a tree is connected it cannot have a vertex of 0 degree. Each vertex contributes at least 1 to the above sum. Thus there must be at least two vertices of degree exactly 1.

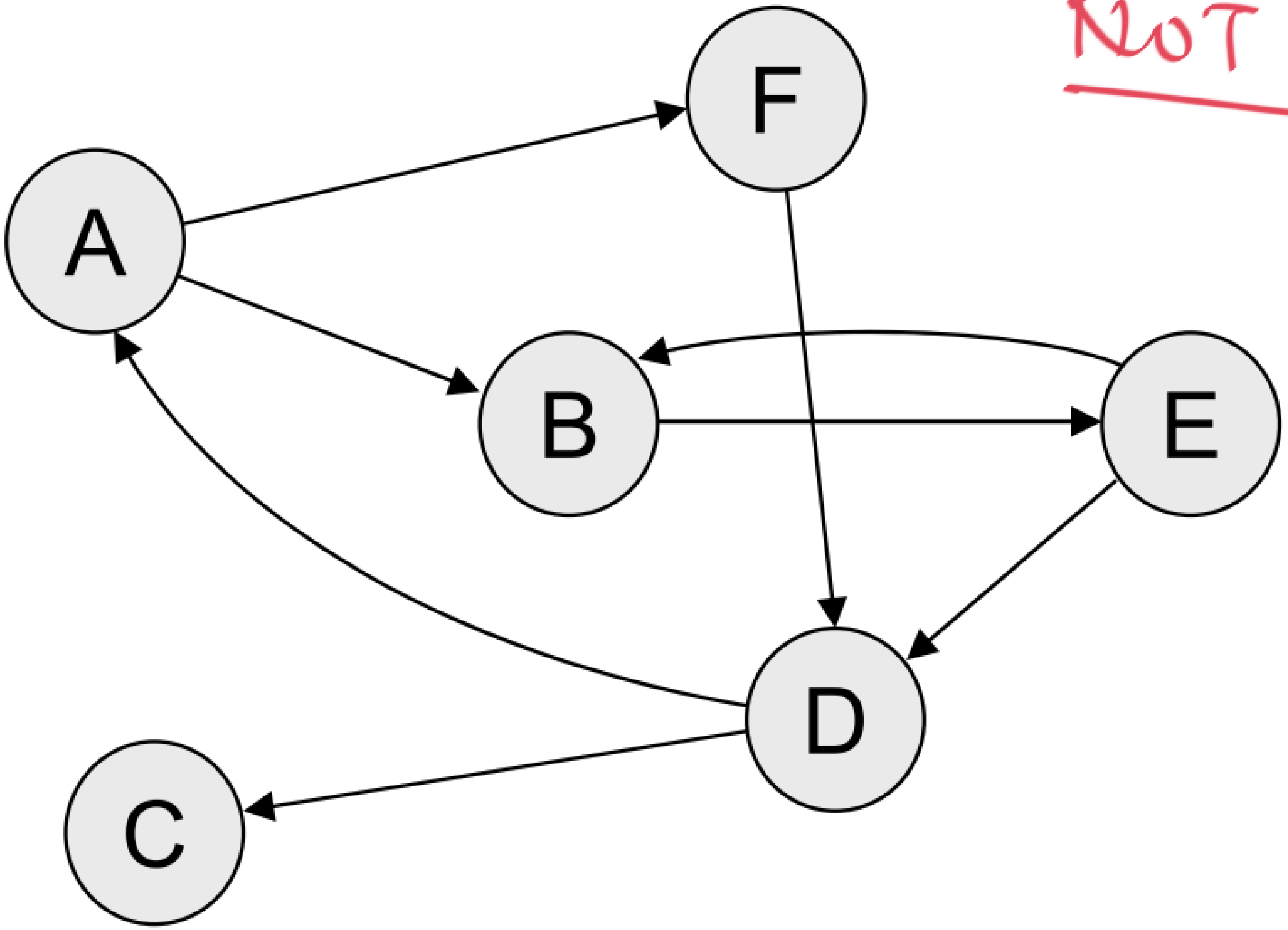
Graph Theory :

Next Topic :

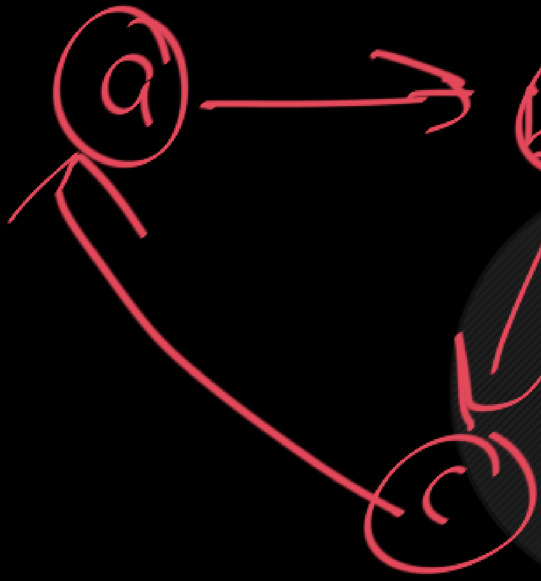
Directed Acyclic Graphs (DAG)

Website : https://www.goclasses.in/

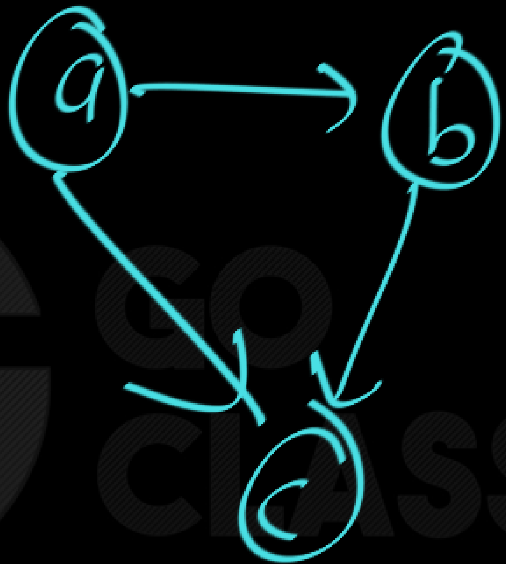
A digraph $G = (V,E)$



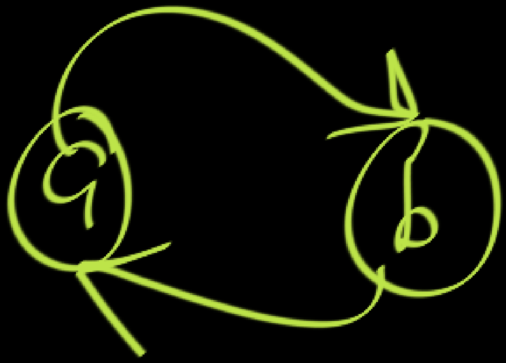
NOT DAG



NOT DAG

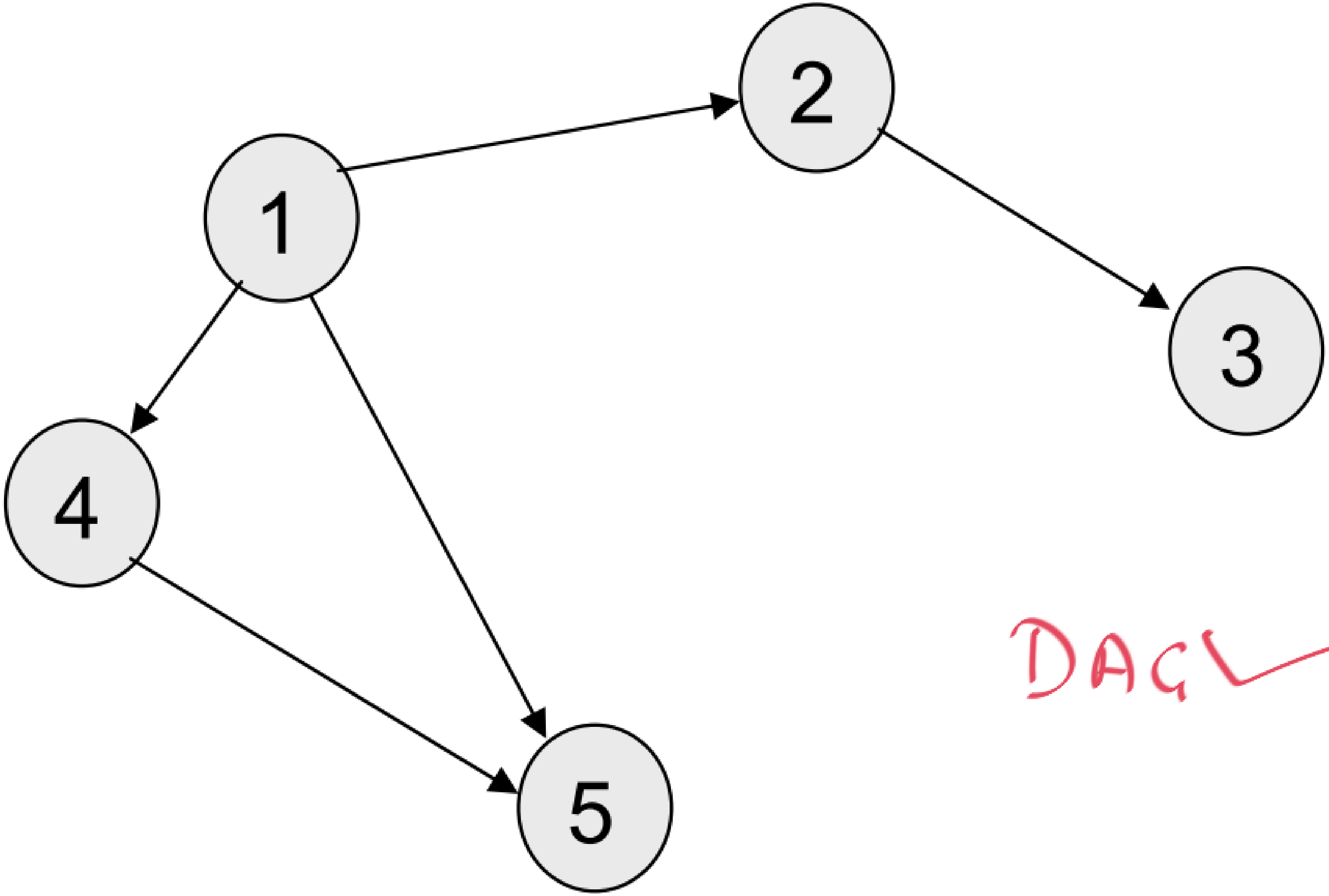


DAG ✓

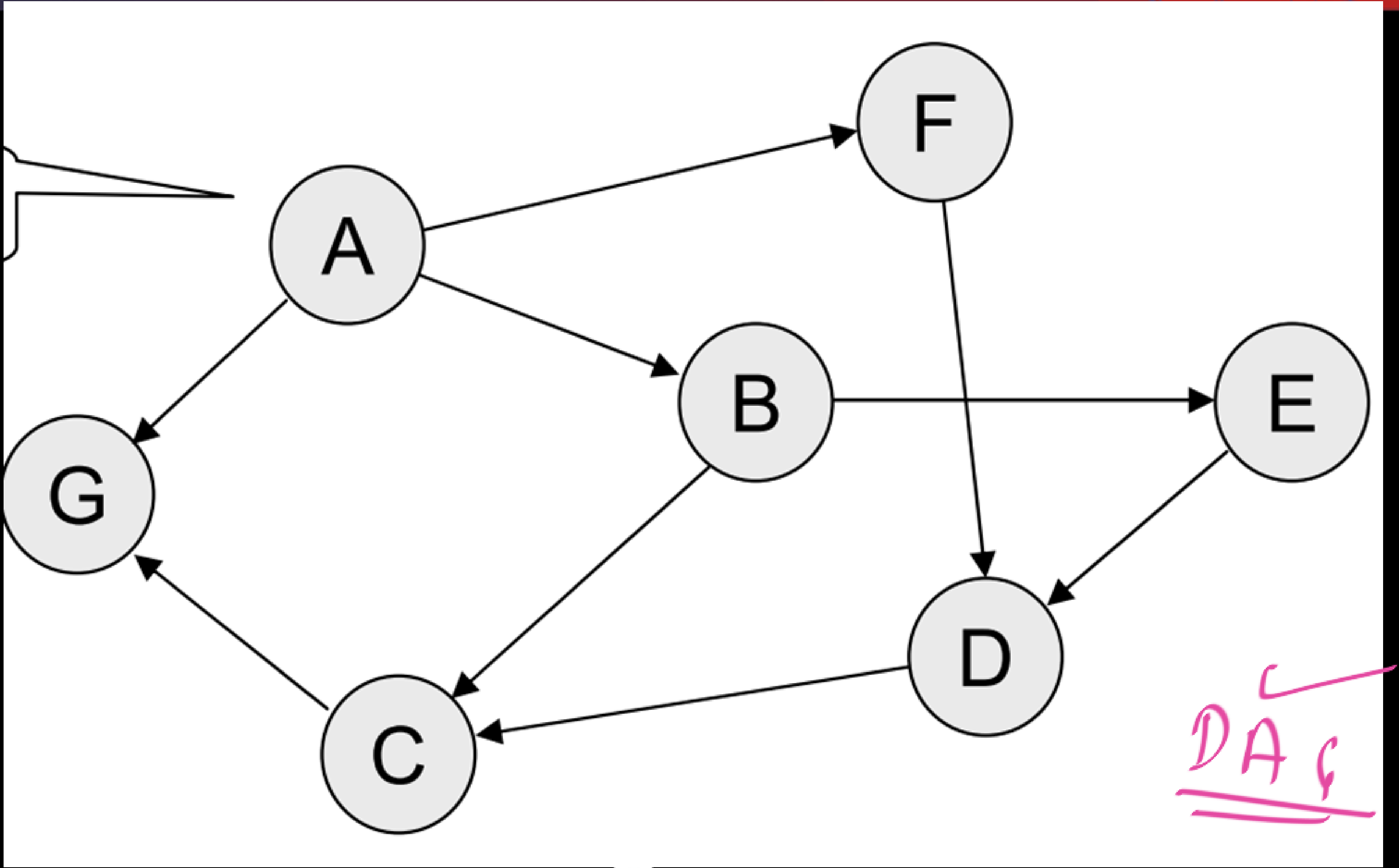


NOT DAG

A directed acyclic graph (or DAG) is a digraph that has no cycles.



DAG ✓


A
F
B
E
G
D
C
DAG

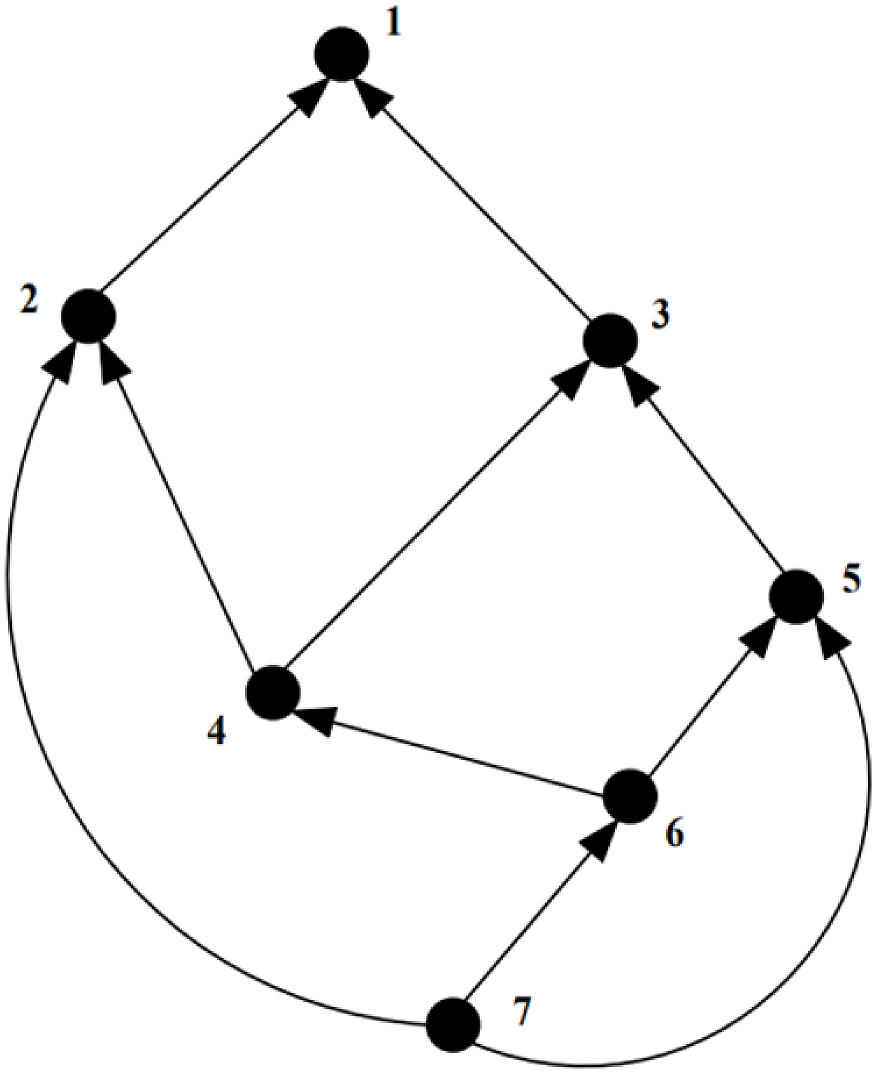


Figure 4: *A directed acyclic graph.*

DAG

Graph Theory :

Next Topic :

Rooted Trees

(Special Types of DAGs) Binary Trees

Rooted Tree = Directed Graph

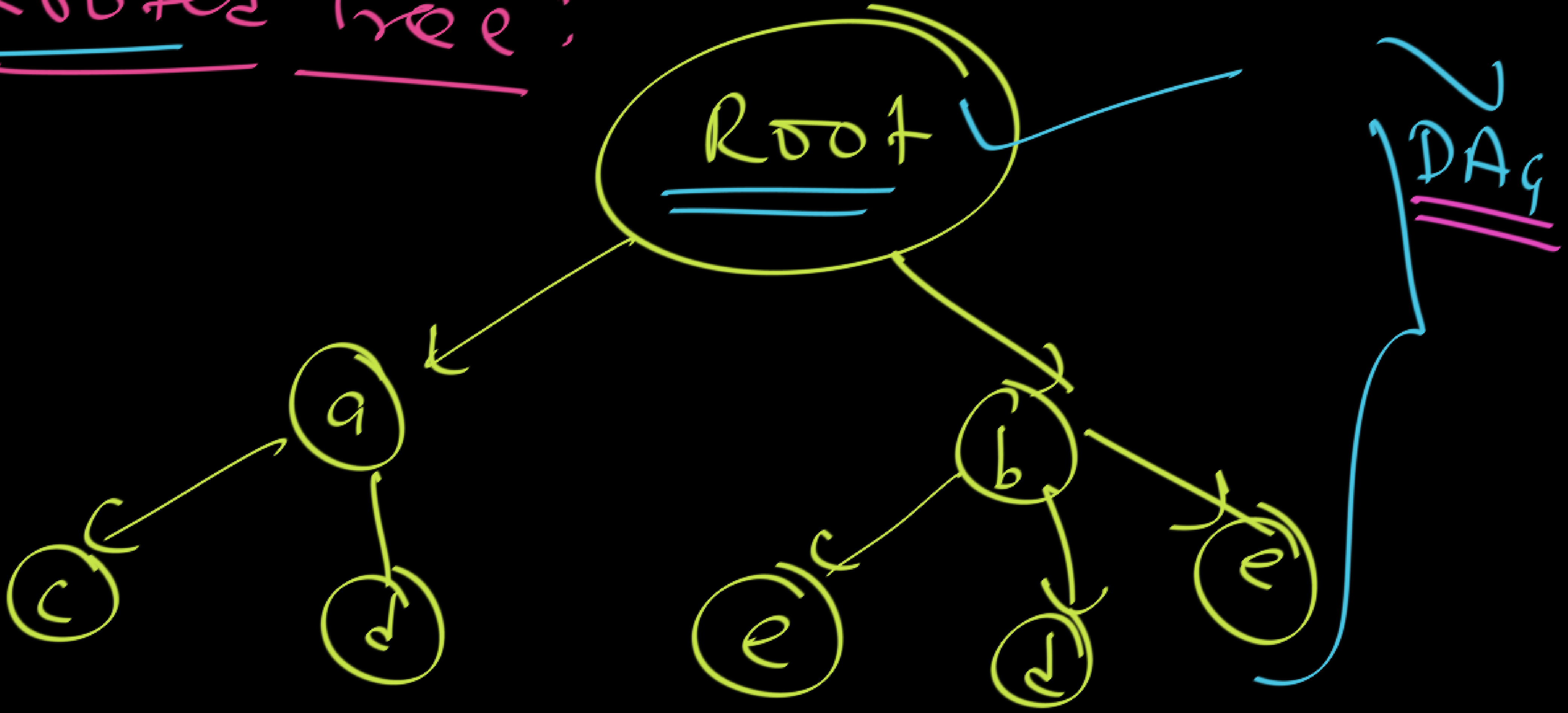
Special type of DAG

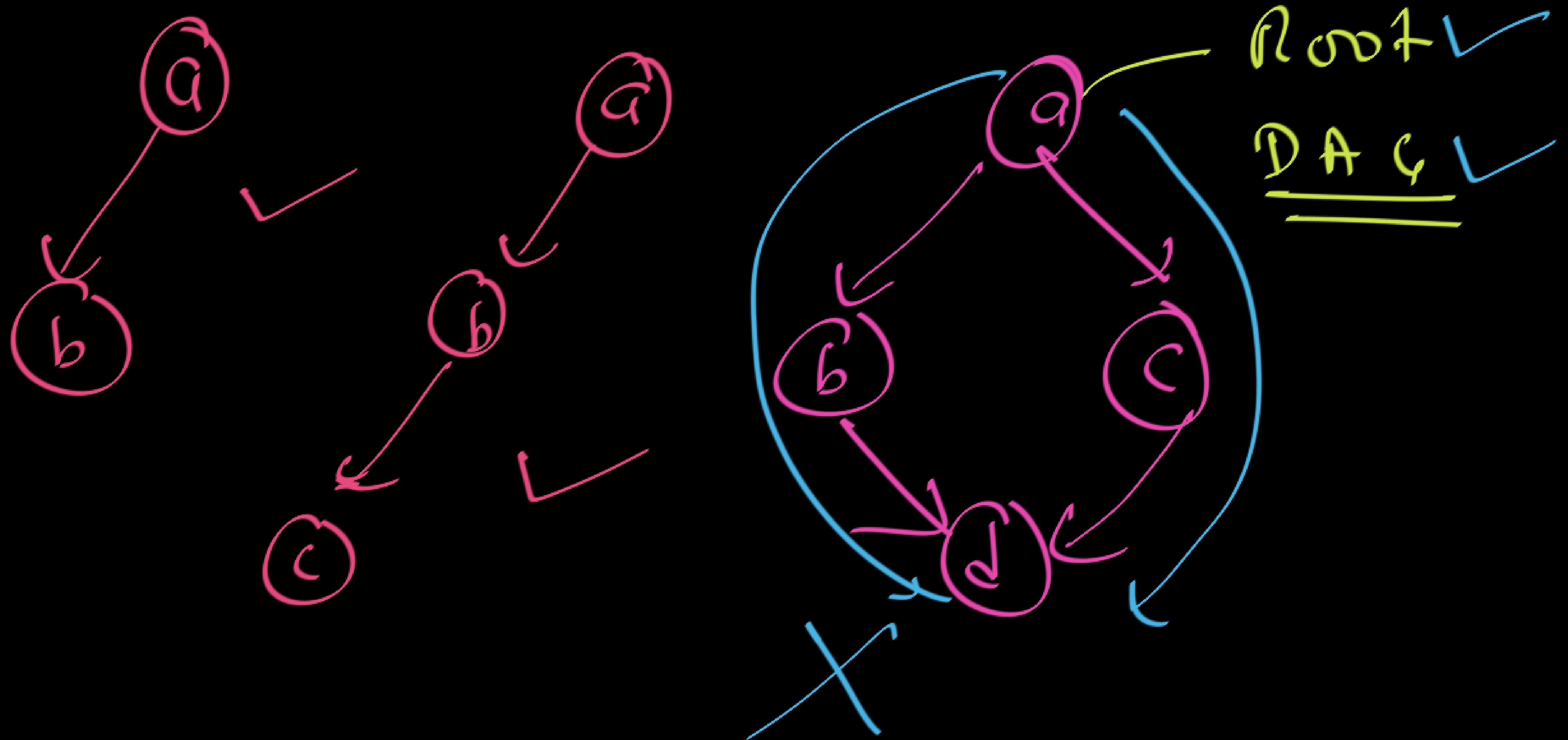
Tree = Undirected Graph

Rooted Tree = misunderstood by majority students

Rooted Tree: $\longrightarrow$ DAG Graph with a Root AND unique Path from Root to every other vertex.

Rooted Tree :

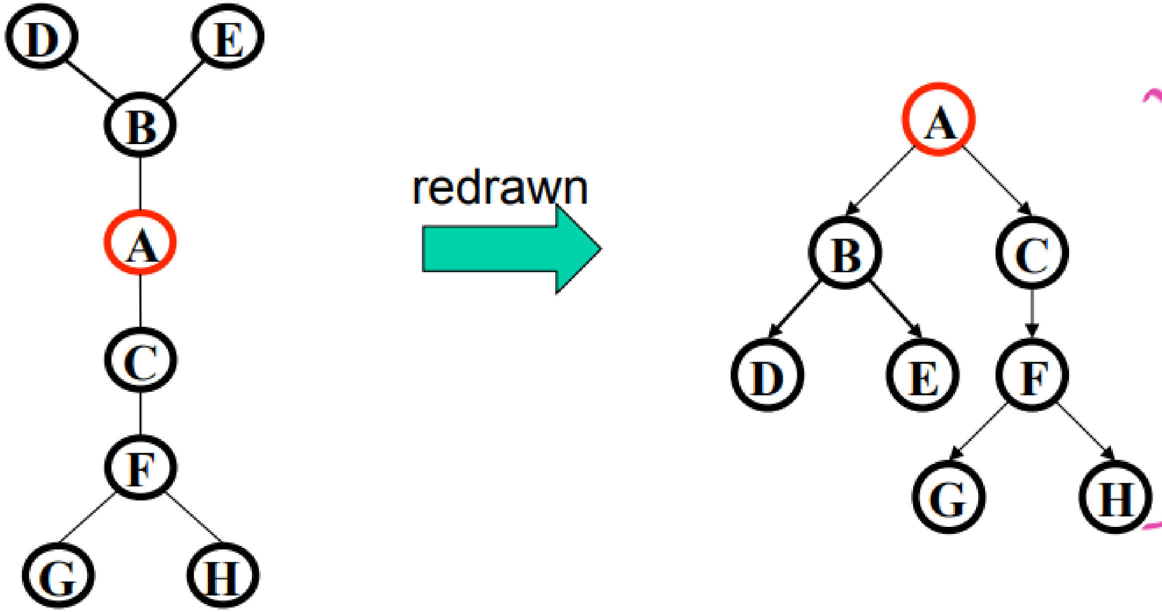



a
b
a
b
c
a
b
c
d
Root
DAG

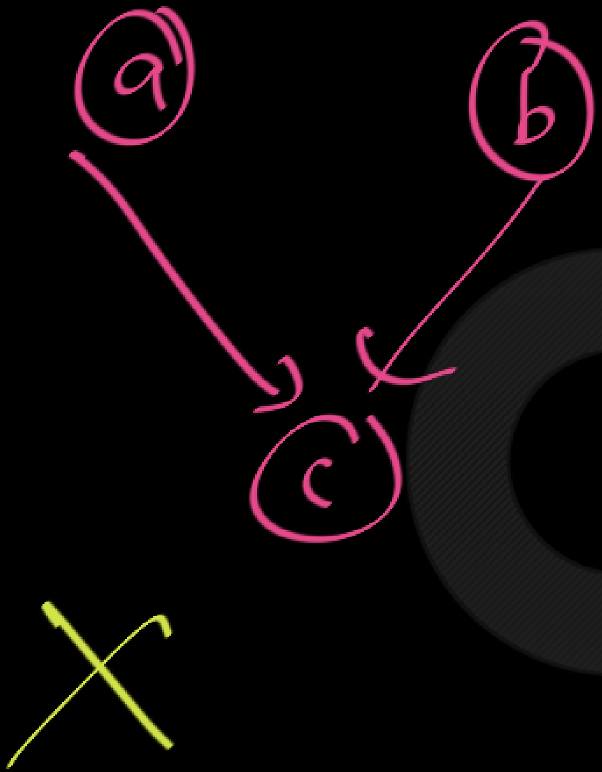
Rooted Trees

- We are more accustomed to rooted trees where:
  - We identify a unique root
  - We think of edges as directed: parent to children
- Given a tree, picking a root gives a unique rooted tree
  - The tree is just drawn differently

Tree
↓
undirected



Rooted Tree
↓
DAG (Special)

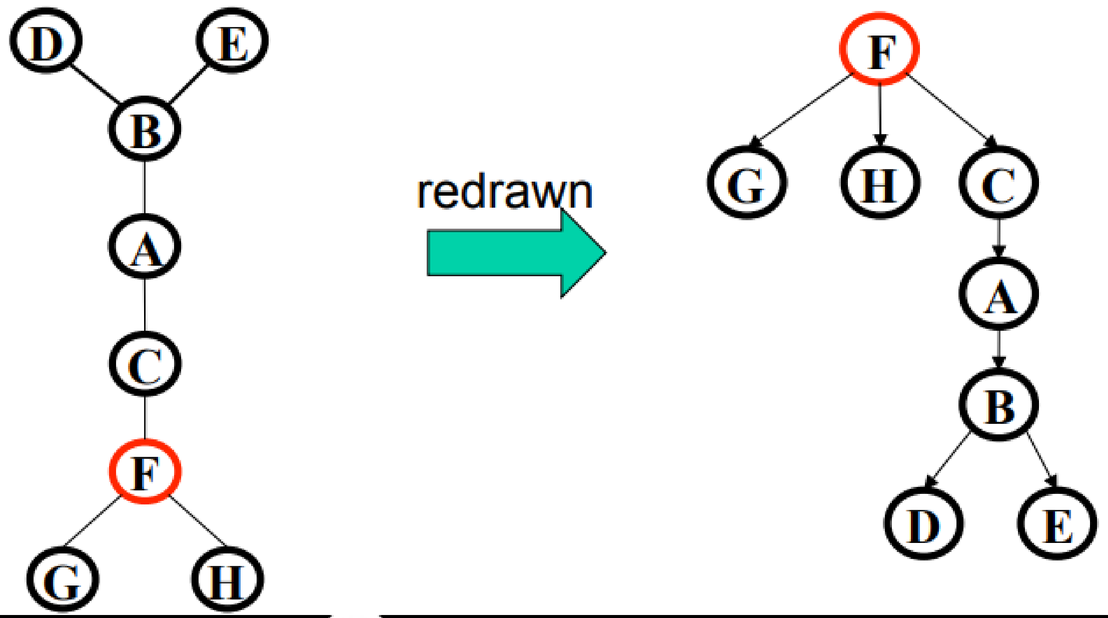

a
b
c

Rootes Tree:

1. Unique Root
2. DAG
3. from Root to every other node, Unique path

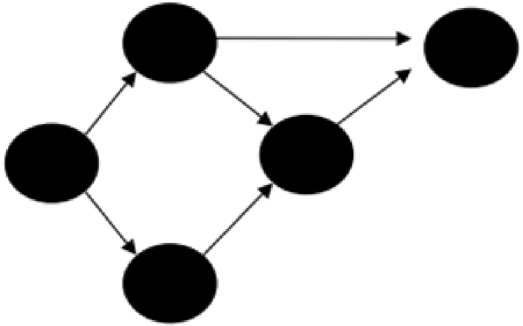
Rooted Trees

- We are more accustomed to rooted trees where:
  - We identify a unique root
  - We think of edges as directed: parent to children
- Given a tree, picking a root gives a unique rooted tree
  - The tree is just drawn differently

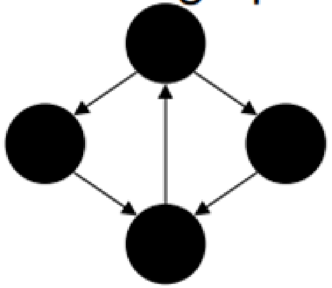


Directed Acyclic Graphs (DAGs)

- A DAG is a directed graph with no (directed) cycles
  - Every rooted directed tree is a DAG
  - But not every DAG is a rooted directed tree

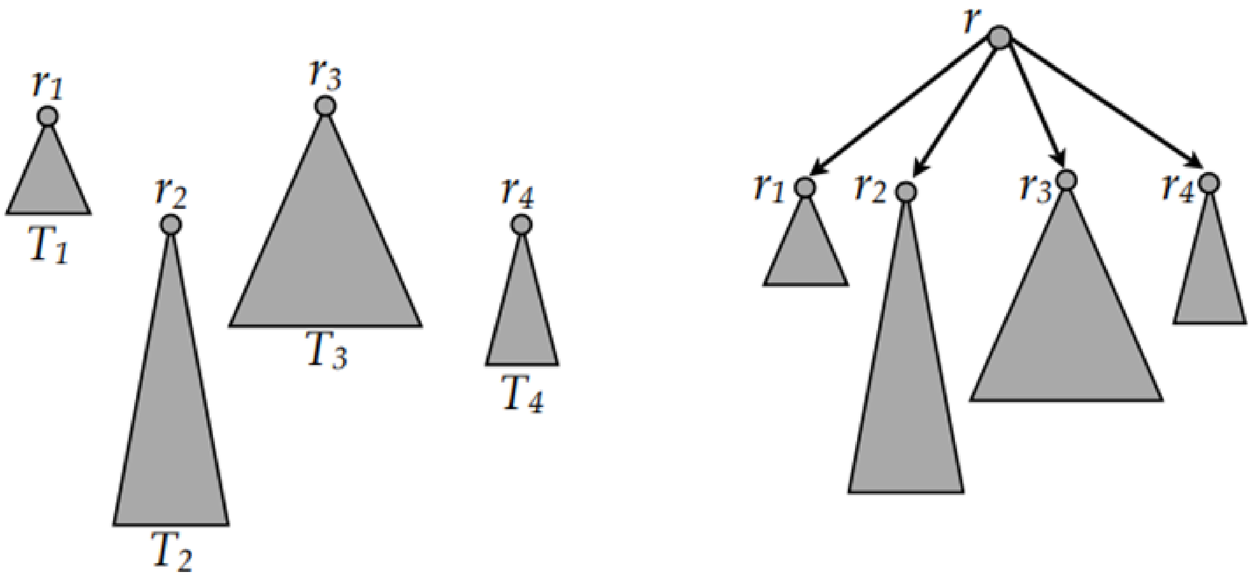


  - Every DAG is a directed graph
  - But not every directed graph is a DAG



Definition – Rooted Tree

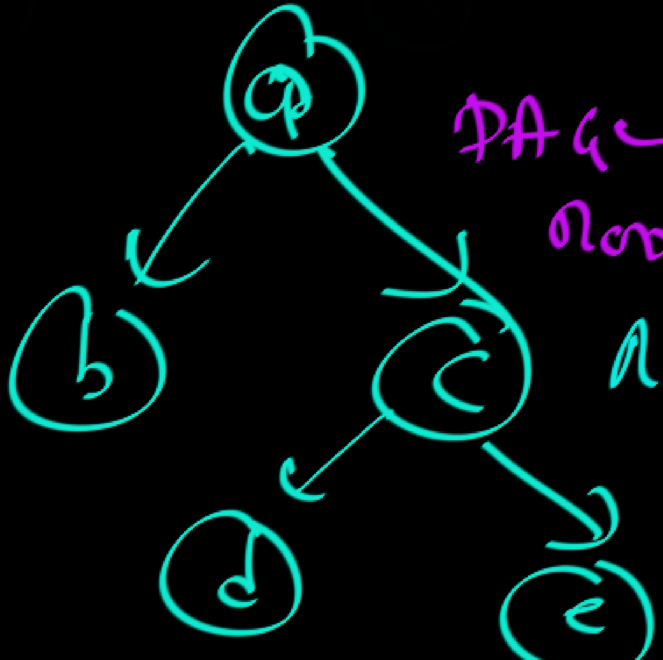
- Λ is a tree
- If $T_1, T_2, \ldots, T_k$ are trees with roots $r_1, r_2, \ldots, r_k$ and r is a node $\notin$ any T_i, then the structure that consists of the T_i, node r, and edges (r, r_i) is also a tree.



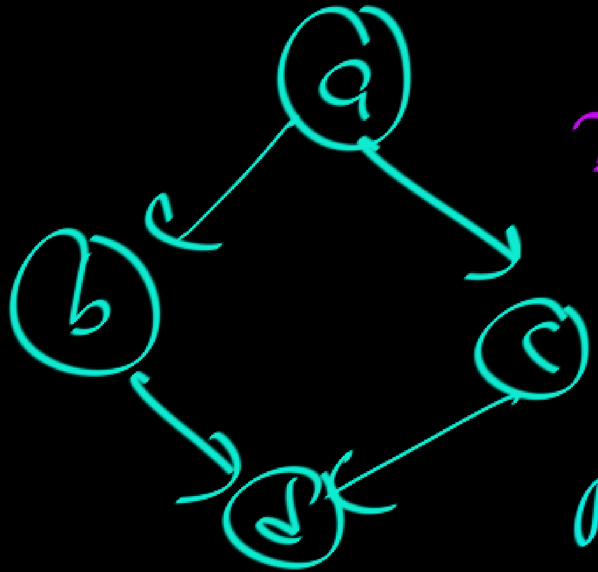
Properties of Rooted Tree:

1. There is a designated Node which is called Root (usually drawn at top)
2. Directed Graph (Direction from Root to other nodes)

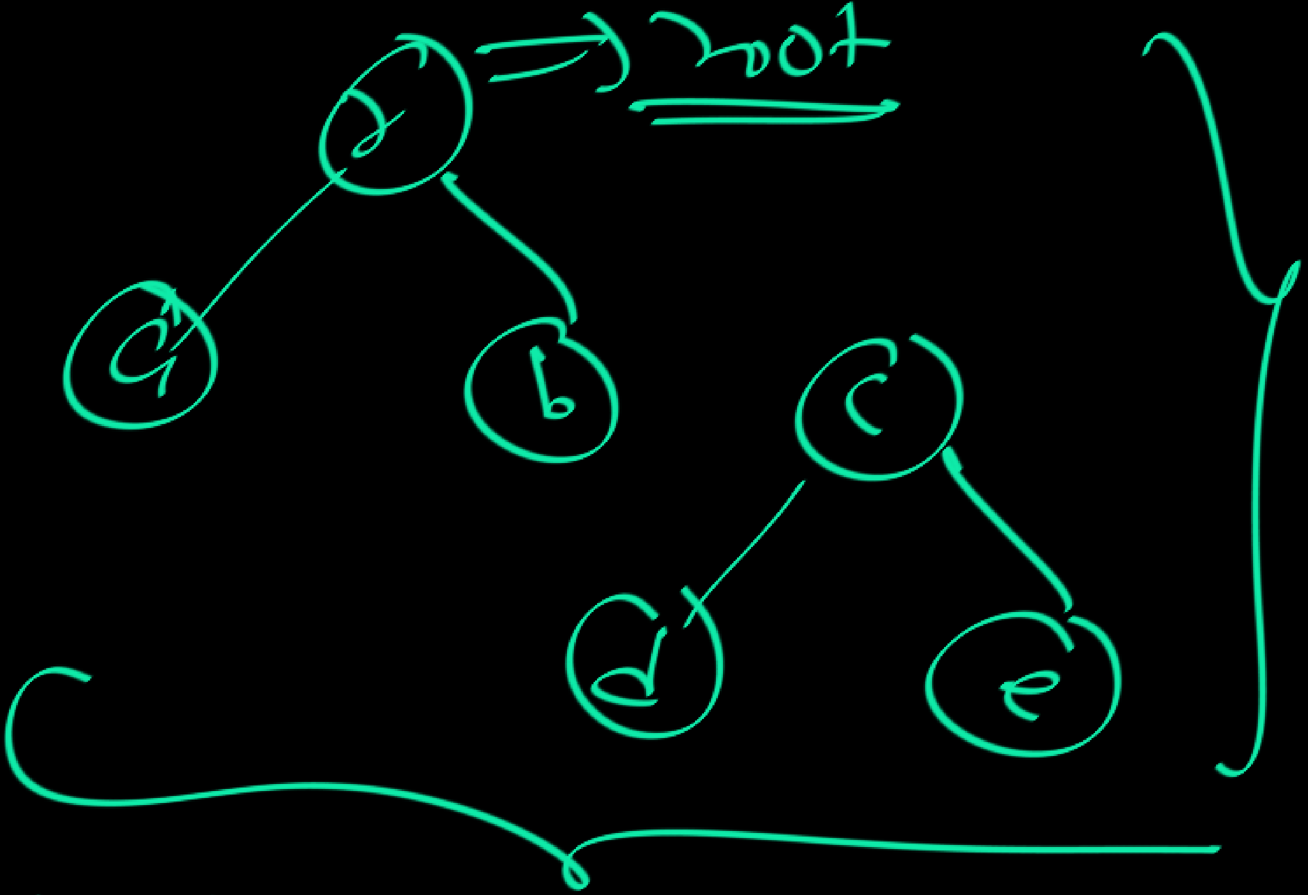
3 from root to any node, there is exactly one (unique) path.



DAG ✓
Root ✓
Rooted Tree

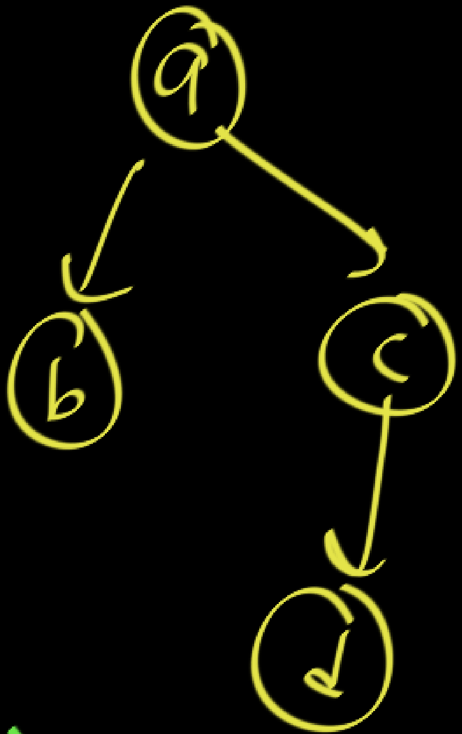


DAG ✓
Root ✓
Not Rooted Tree



Not rooted Tree

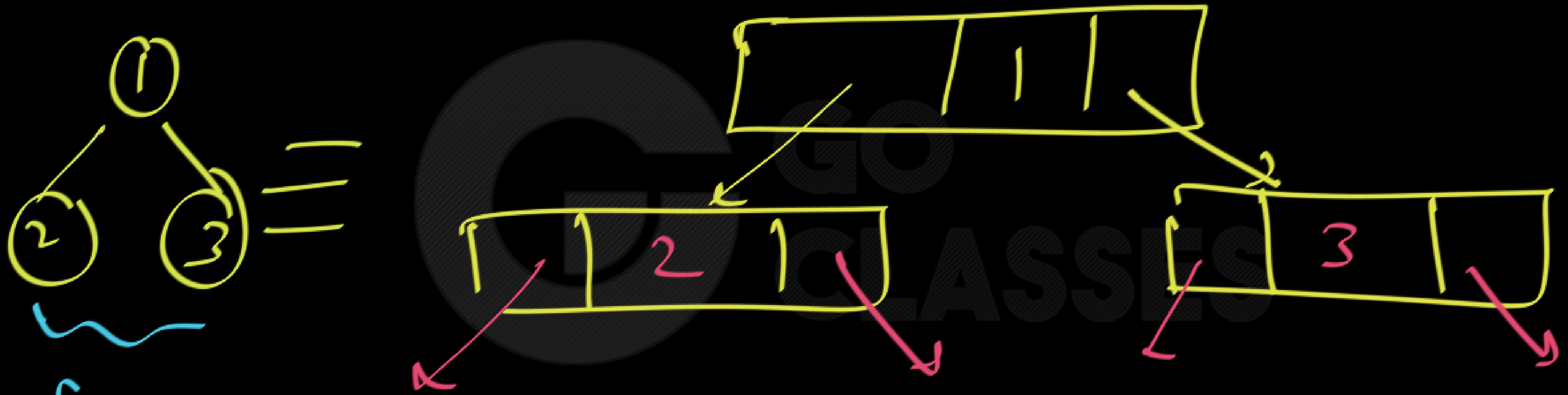
from root to c There is No Path, So, Not Rooted Tree



$\equiv$

for Convenience

Analogy: → Binary Tree Data Structure

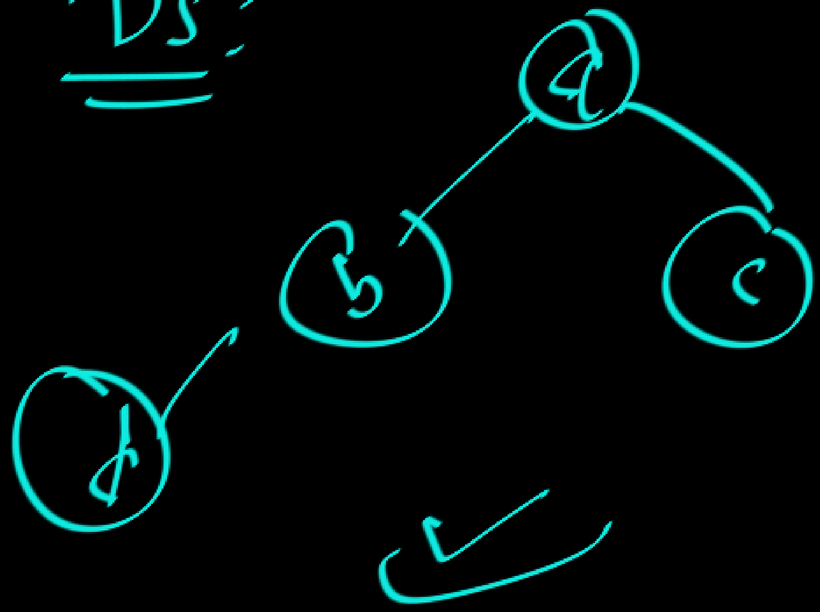


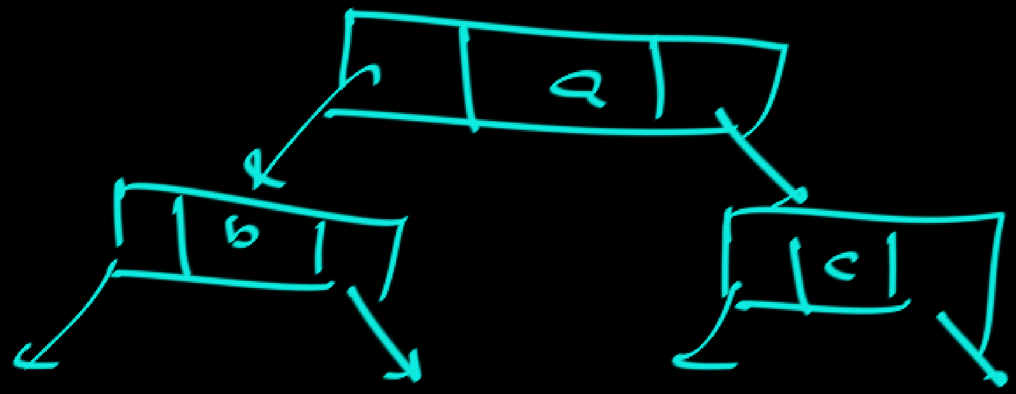
for Convenience

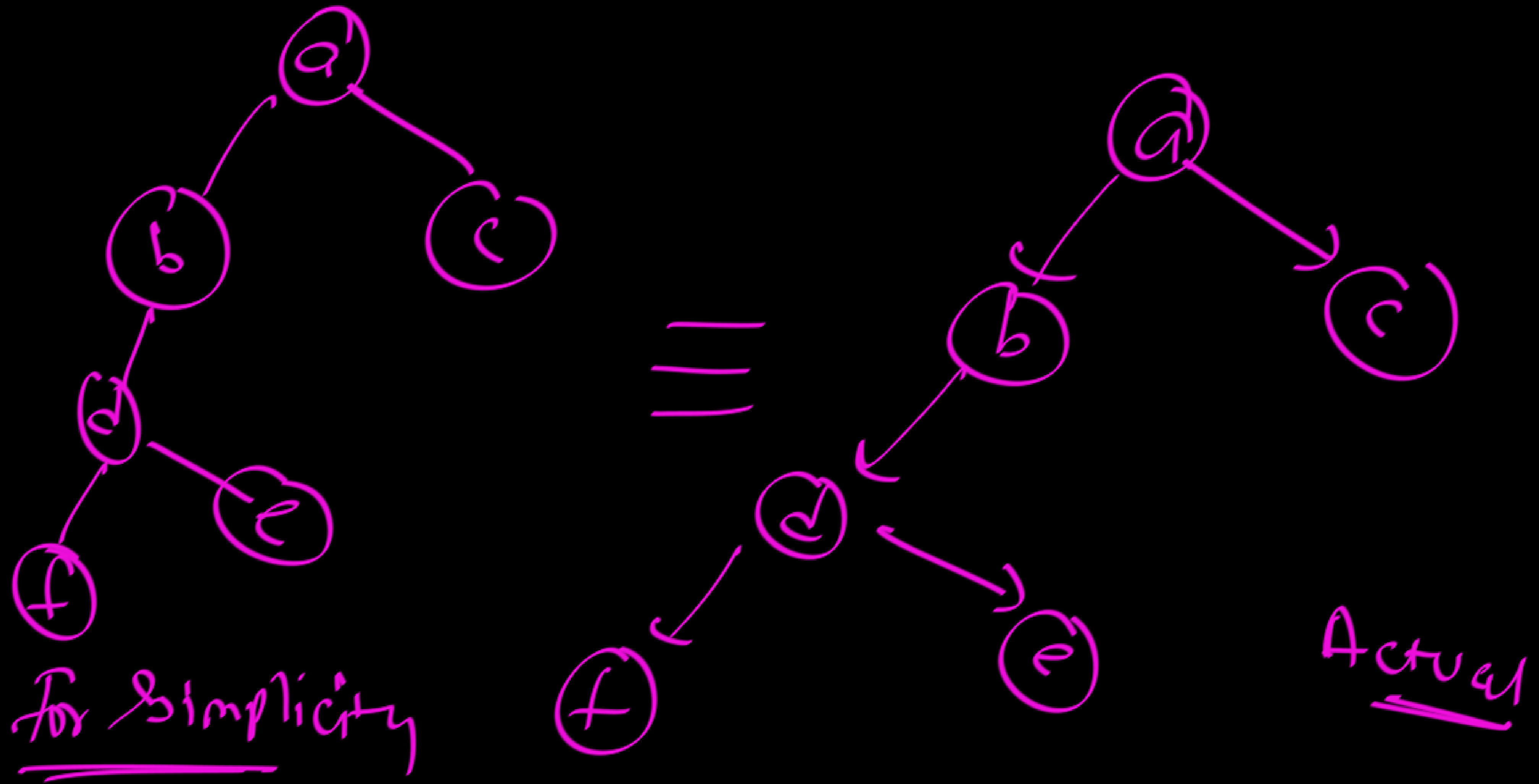
Actual thing

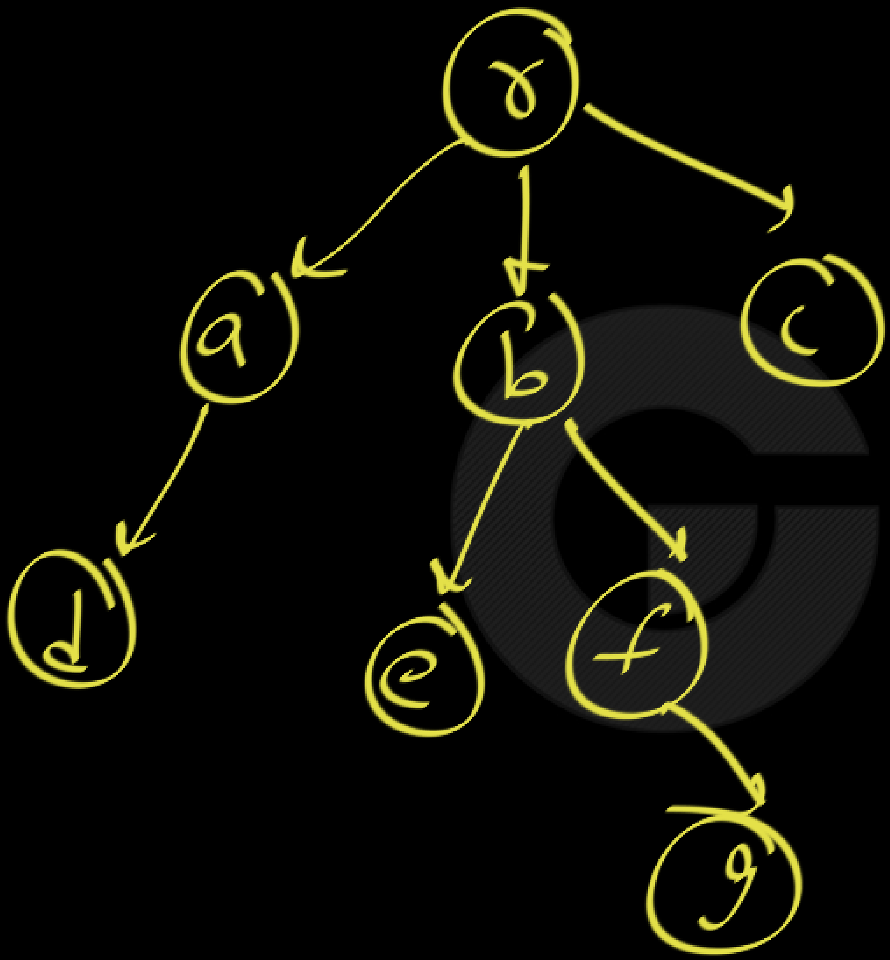
Rootes Trees, for Convinience, are Drawn without Directions.

DS:






a
b
c
d
e
f
for simplicity
≡
a
b
c
d
e
f
Actual

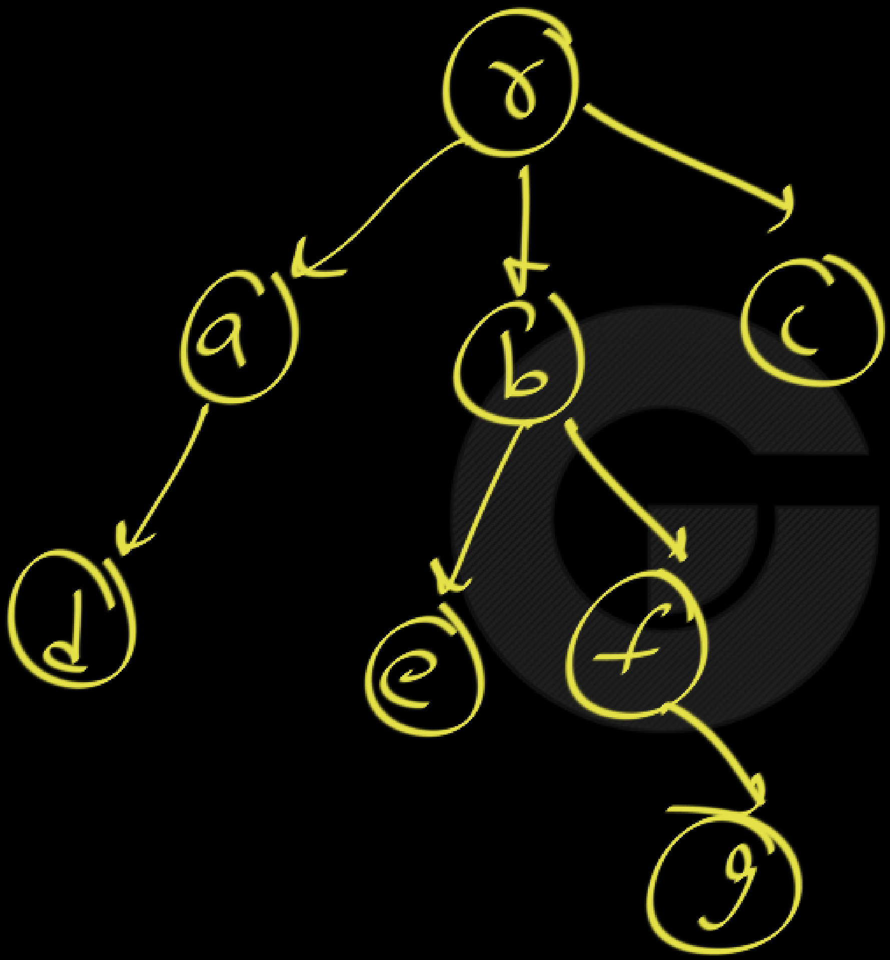


Root = r ✓

Children of Root = {a, b, c}

Children of b = {e, f}

Parent of c = r



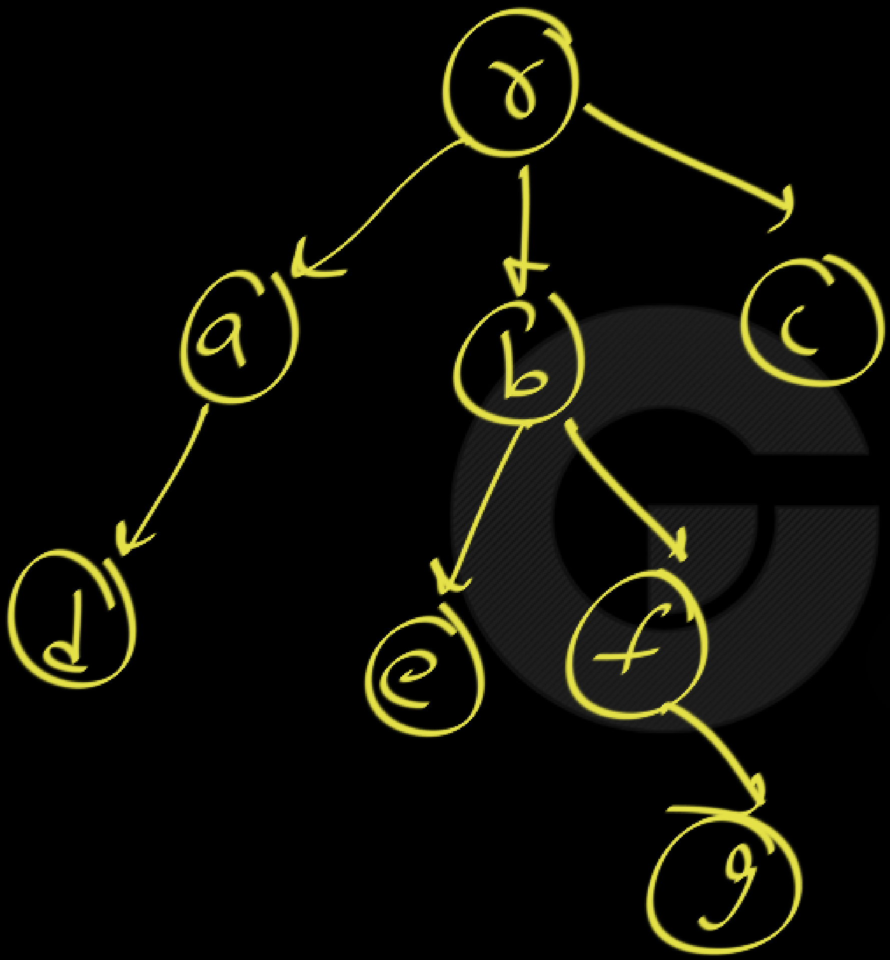
Parent of g = f

Siblings of b = {a, c}

Siblings of f = {e}

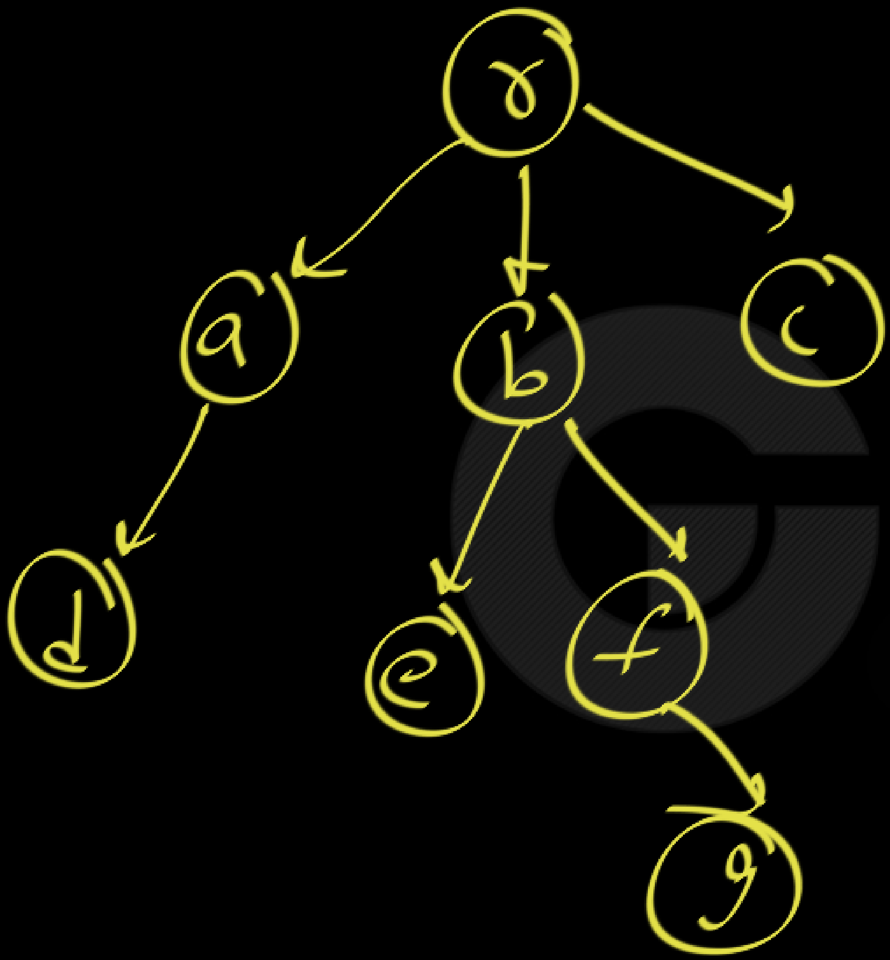
Siblings of d = None

Parent of c = r



Ancestors of c $= \{r\}$

Ancestors $(g) = \{f, b, r\}$



Path from r to f:

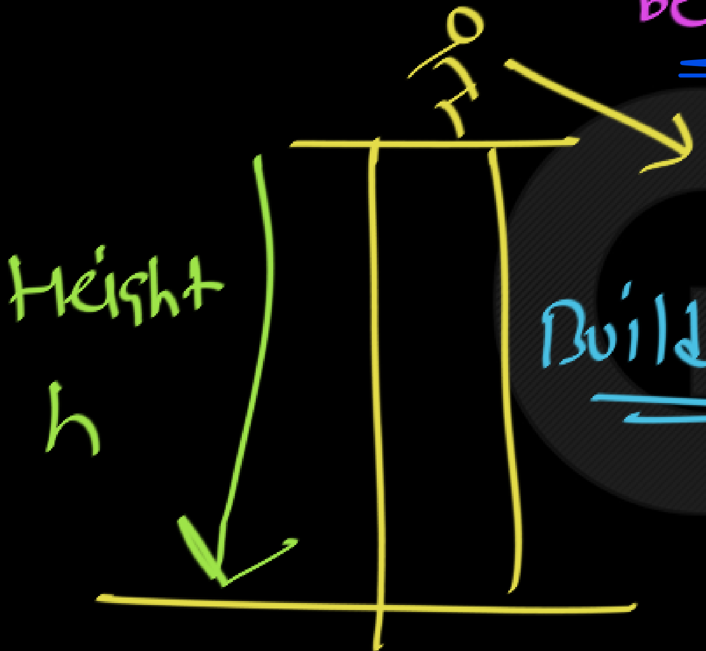
$r \rightarrow b \rightarrow f$

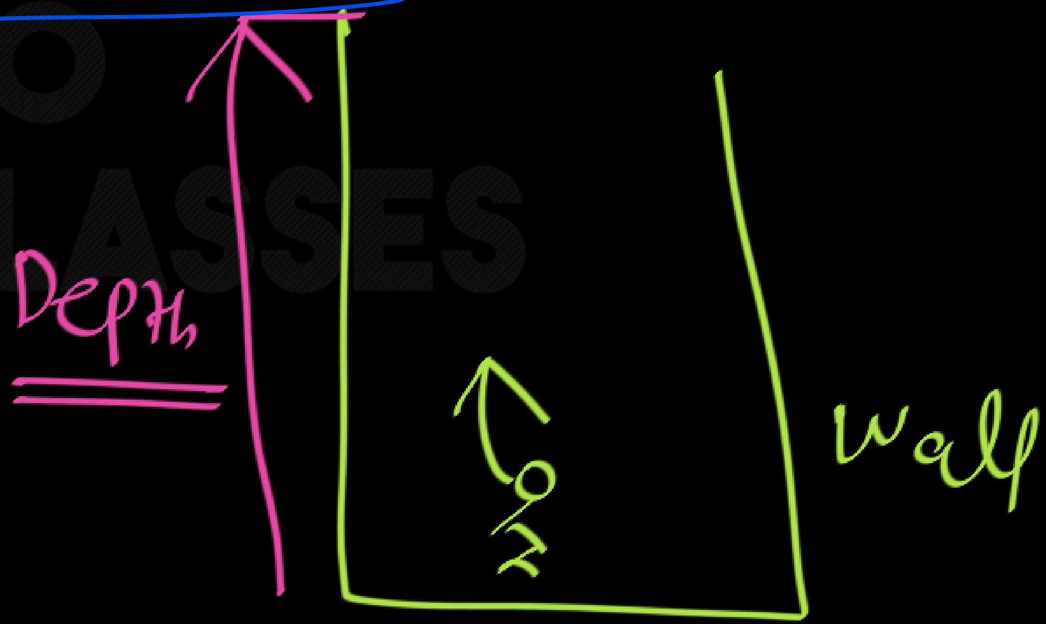
Path from b to g:

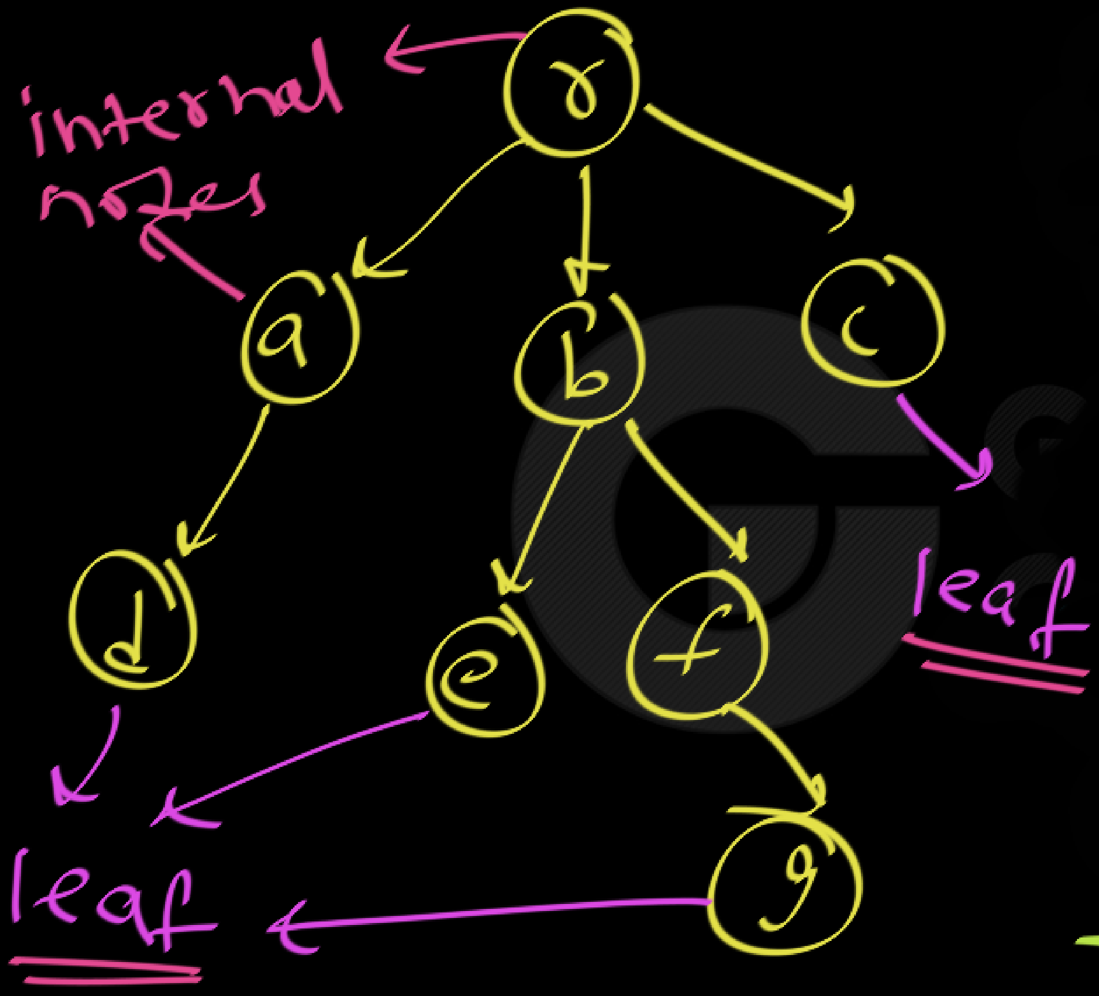
$b \rightarrow f \rightarrow g$

Height := #Edges below

Depth = #Edges Above







	height	Depth
r	3	0
a	1	1
f	1	2
g	0	3
d	0	2

Height of a node x in Rooted Tree:

$\underbrace{\text{from } x \text{ to the farthest leaf}}$

Edges

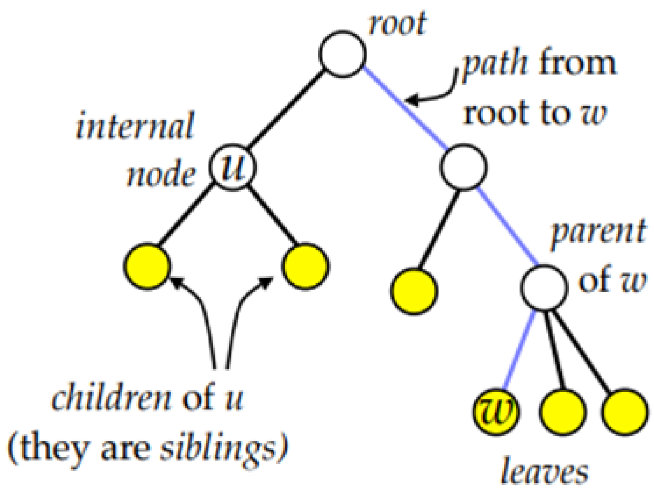
Depth of a node x in Rooted Tree :

$\underbrace{\text{from Root to } x}$

#Edges

Terminology

- r is the *parent* of its *children* $r_1, r_2, \ldots, r_k$.
- $r_1, r_2, \ldots, r_k$ are *siblings*.
- *root* = distinguished node, usually drawn at top. Has no parent.
- If all children of a node are Λ, the node is a *leaf*. Otherwise, the node is a *internal node*.
- A *path* in the tree is a sequence of nodes $u_1, u_2, \ldots, u_m$ such that each of the edges (u, u_{i+1}) exists.
- A node u is an *ancestor* of v if there is a path from u to v.
- A node u is a *descendant* of v if there is a path from v to u.



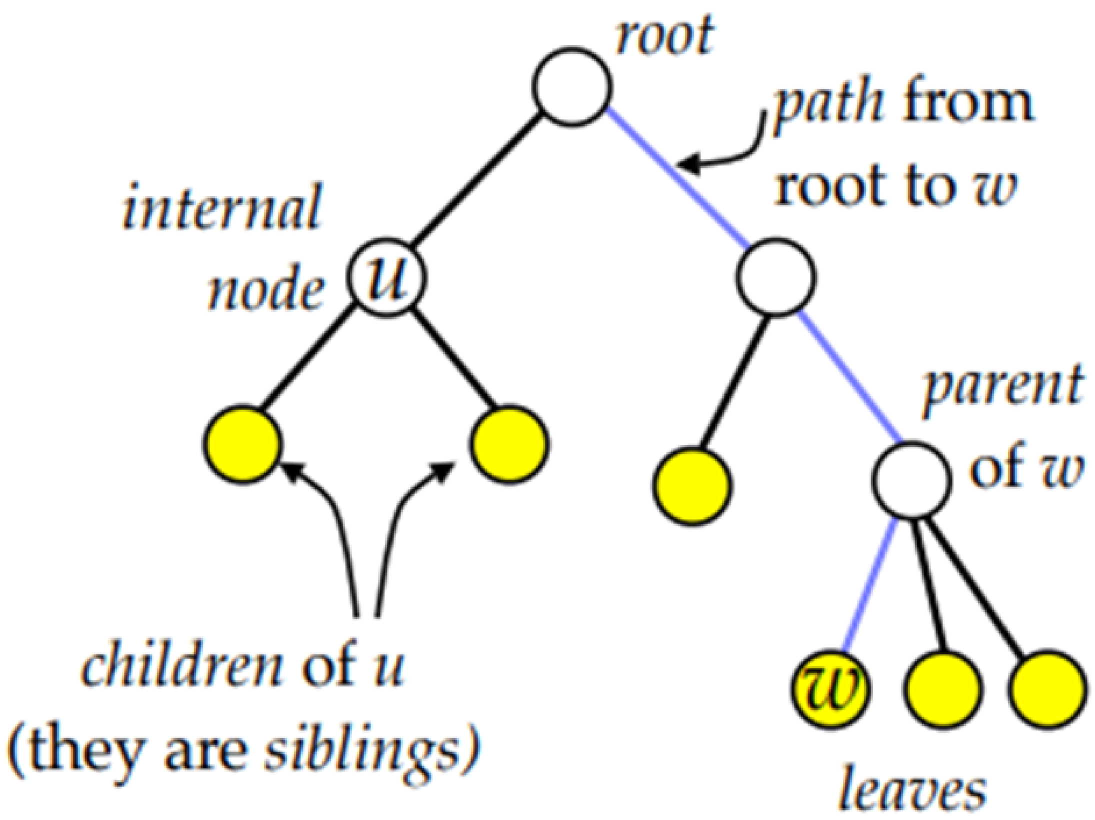
Leaf Node: node without child

Internal Node: node with at least one child.

G: a — Root ✓

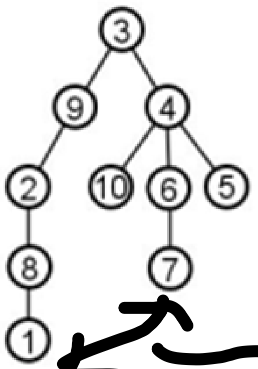
Leaf ✓

NOT internal node


root
path from root to w
internal node
u
parent of w
w
children of u (they are *siblings*)
leaves

- Vertices with the same parent are called siblings

Questions about Rooted Trees



- Suppose that vertices u and v are siblings in a rooted tree.
- Which statements about u and v are true?

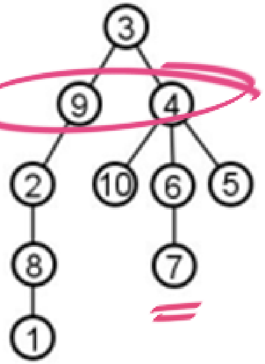
1. They must have the same ancestors
2. They can have a common descendant
3. If u is a leaf, then v must also be a leaf

- Vertices with the same parent are called siblings

Questions about Rooted Trees

Same Parent

Siblings



- Suppose that vertices u and v are siblings in a rooted tree.
- Which statements about u and v are true?

1. They must have the same ancestors
2. They can have a common descendant
3. If u is a leaf, then v must also be a leaf

at least one

Tree

Undirected Graph

NOT Applicable Here

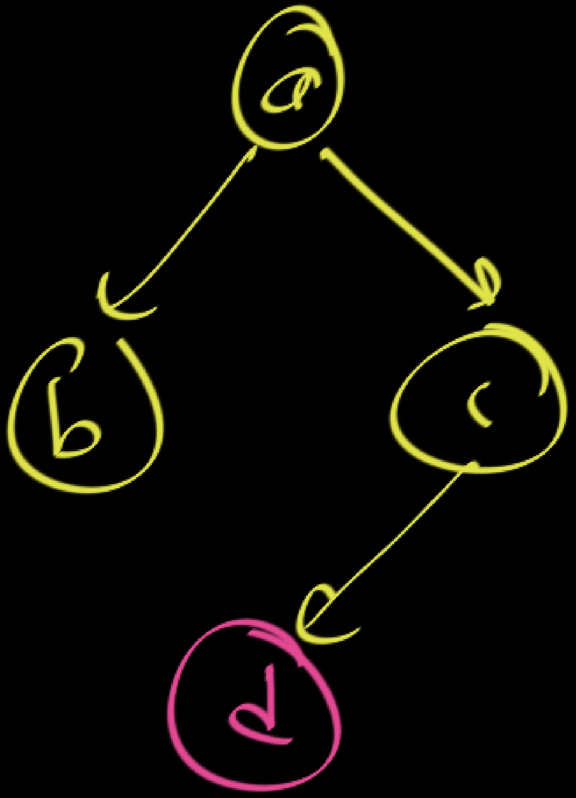
Rooted Tree

Directed Graph

Concepts : Root, internal node, leaf, Height, Depth, siblings

Rooted Tree is not a Tree in Graph theory.

But we think of it as a Tree sometimes.



Sometimes

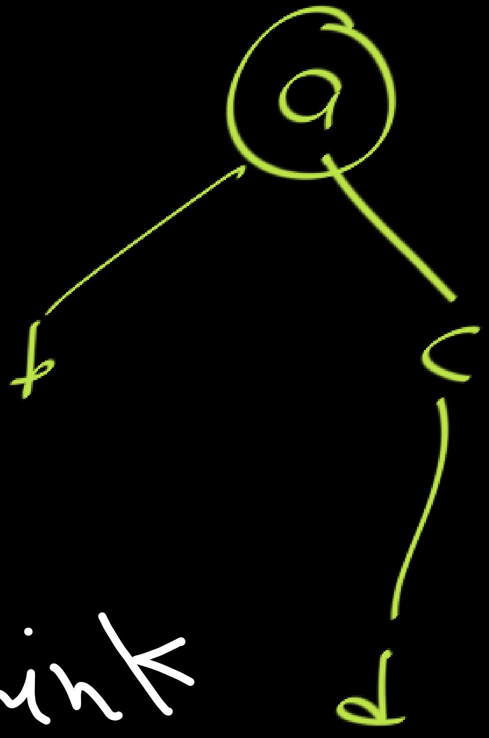
thought

~~as~~

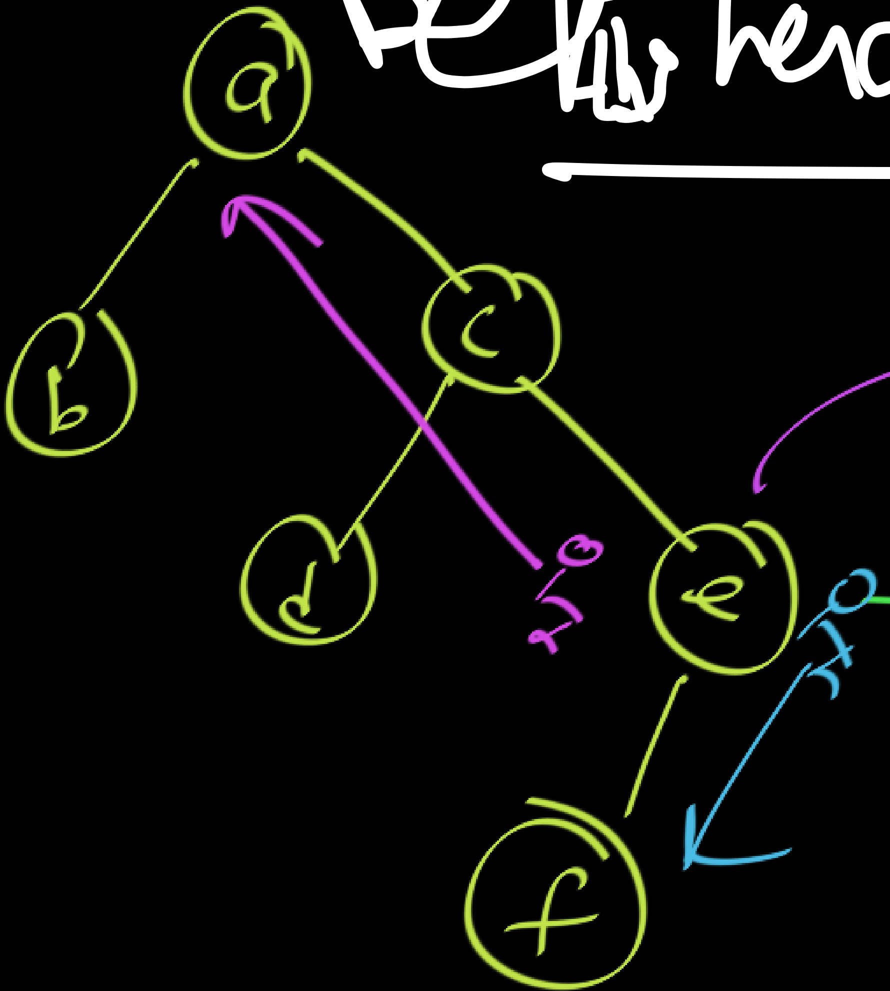
think

a

Tree



Depth height

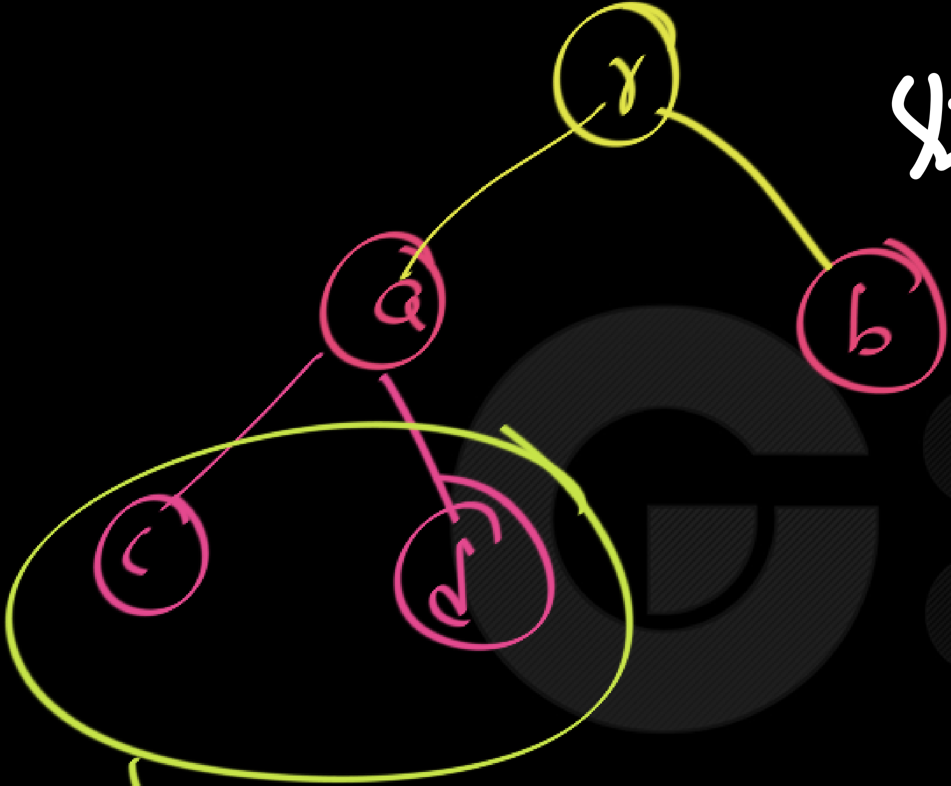


Root to e

Depth of e = 2

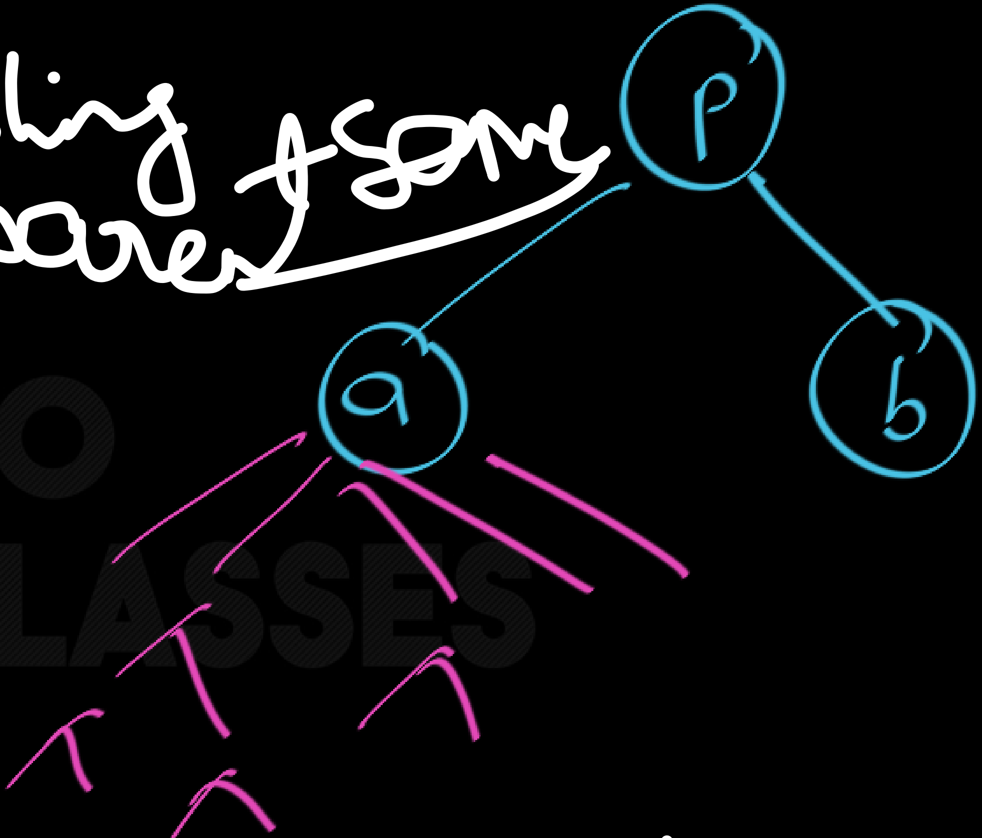
Height of e = 1

e to farthest Leaf



Descendents of a

Sibling: parent same



Same lvl/ kuc

Q: Given Tree T, $n \geq 2$ vertices $= n$

Proof

$a_1 \geq 1$ $\boxed{a_2 \geq 1 \quad a_3 \geq 1 \quad a_4 \geq 1 \quad a_5 \geq 1}$ $n-1$ Vert.

$\to$ Deg ≥ 2

proof atleast 2 vertices hae deg 1

deg $= 1$

tree connected , so vertices degree more than one

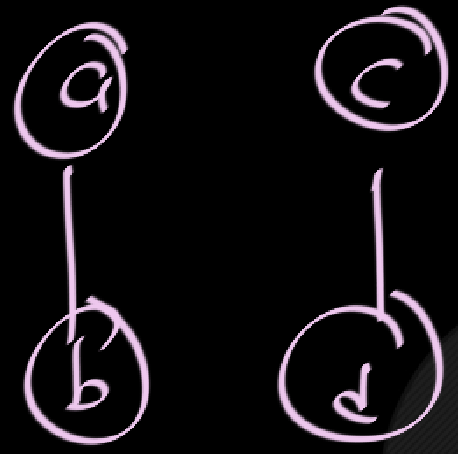
let assume dont have atkeast two of vertices of deg one means atmost deg 1 one vetices

$\boxed{\text{Total Deg} = (n-1)2 + 1}$ Contradiction with $(n-1)2$

so u have the atleast 2 veteices of deg one another name

forest ≡ Acyclic ✓

Tree ≡ Connected, Acyclic



Tree ✗ not connected

Acyclic ✓ ≡ forest

every tree is a forest , not in vive versa

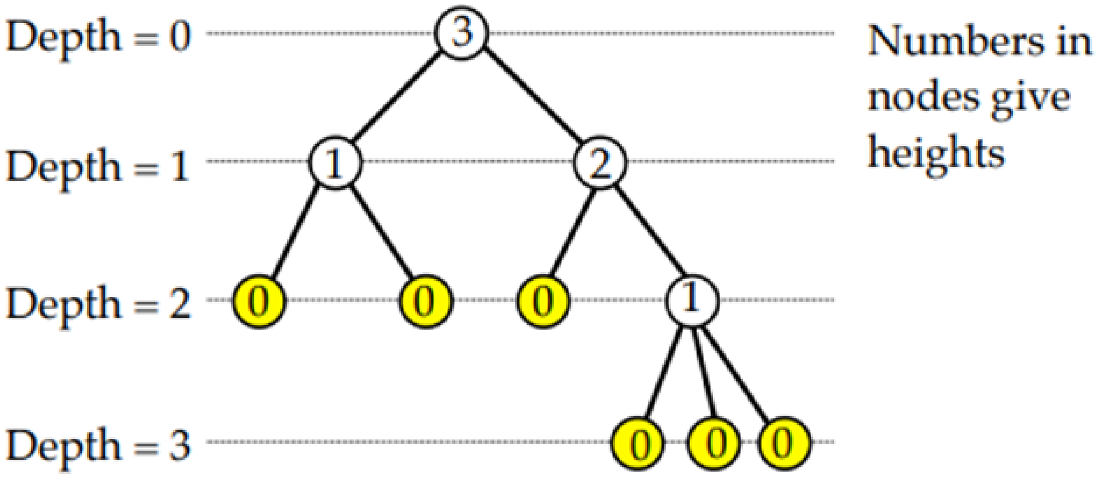
forest connected is a tree

Rooted trees

- A *rooted tree* is a tree with a designated vertex v_0 called the *root*. According to Theorem 30, each vertex v_i is connected to the root by a unique path $v_i, v_{i-1}, \ldots, v_1, v_0$, in which case v_{i-1} is called *father* (or *parent*) of v_i, and v_i is a *child* (or *son*) of v_{i-1}.
- The *height* of a rooted tree is the distance from the root to the farthest leaf.
- A *binary* tree is a rooted tree in which every vertex has at most two children.
- A *full binary tree* is a binary tree in which every non-leaf vertex has exactly two children.

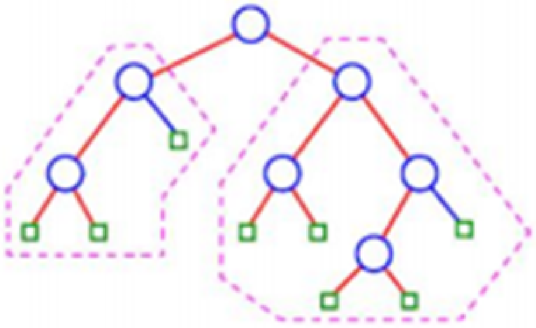
Height & Depth

- The *height* of node u is the length of the longest path from u to a leaf.
- The *depth* of node u is the length of the path from the root to u.
- Height of the tree = maximum depth of its nodes.
- A *level* is the set of all nodes at the same depth.



Subtrees

- Given a rooted tree and a node v, the subtree rooted at v includes v and its descendants.



Basic Properties

- Every node except the root has exactly one parent.
- A tree with n nodes has n-1 edges (every node except the root has an edge to its parent).
- There is exactly one path from the root to each node.

- **DEFINITION:** Nodes with no children are called *leaf* nodes. All other nodes are called *interior* nodes.

- **DEFINITION:** The *degree* of a node is the number of its children. The degree of a tree is the maximum degree of any of its nodes.

- **DEFINITION:** Nodes with the same parent are called *siblings*.

- Note that the edges are **directed** edges. This means there is a direction associated with them. This definition has the edges directed to the root. Nobody draws the edges as directed edges, it's just understood.