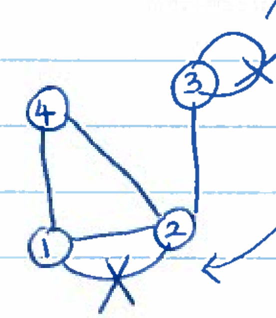
→ Recall : Path from u to v is a sequence of vertices $(v_0, v_1, \ldots, v_k)$ such that $v_0 = u$, $v_k = v$ and $\forall\, i = 1, \ldots, k$

- v is reachable from u if $\exists$ a path from u to v
- Note : Vertex is always ~~fro~~ reachable from itself

→ Undirected $(v_{i-1}, v_i) \in E$

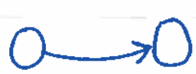


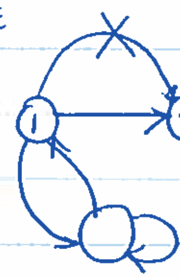
$E = \{ \ldots, \{3,3\}, \ldots \}$ ↳ can't repeat in a set

No self loops

$E = \{ \ldots\ldots, \{1,2\}, \{1,2\}, \ldots \}$ can't repeat in set

Directed $\{v_{i-1}, v_i\} \in E$





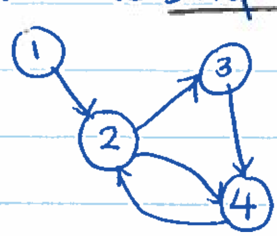
$E = \{ \ldots, \{1,2\}, \{1,2\}, \ldots \}$

Can't repeat element

Self loops are allowed.

- Think in terms of how a set can be formulated.
- If you can't think of the choices for a graph, think about the underlying representation (sets)

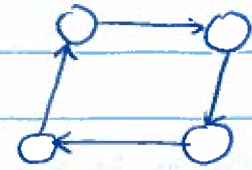
→ Path is simple if all of its vertices are different



$(1, 2, 4)$ → SIMPLE!

$(1, 2, 3, 4, 2, 3, 4)$ → NOT SIMPLE!

→ <u>Directed Case</u>:



$\boxed{\text{cycle}}$

Starts at a vertex and ends back there.

Smallest cycle →  self-loop

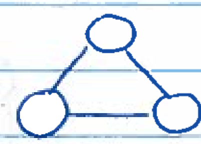
→ <u>Cycle</u>: A path forms a cycle if $v_0 = v_k$ and $\underbrace{k \geq 1}_{\text{at least 1 edge}}$

For a directed graph

and all vertices other than v_0, v_k are distinct

↳ simple cycle

→ What is a cycle in an undirected case?

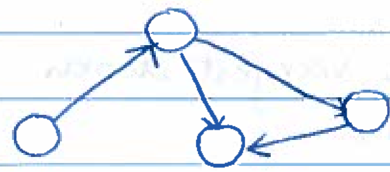
- Same as that of directed? What is the problem?
  - Smallest cycle is 

→ cycle for an undirected graph: Same but $\underbrace{k \geq 3}_{\text{at least 3 edges}}$

→ Graph is <u>acyclic</u> if $\nexists$ cycle.

→ SPECIAL TYPES OF GRAPHS:

→ DAG : Directed Acyclic Graph



Application:
- Course Pre-requisites

(A) → (B) A is required for B

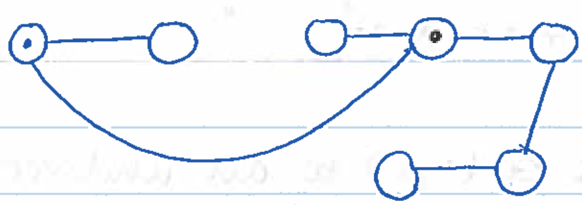
If there is a cycle what does it mean?
- can't resolve dependency

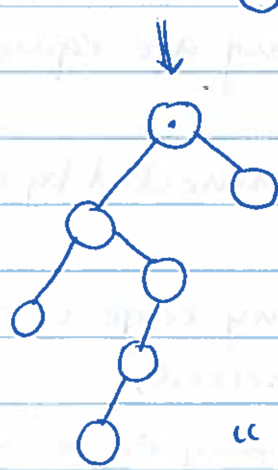
Another application is loading of packages.

? Given a graph, how can you tell if it is acyclic or not? → Will see this later on.

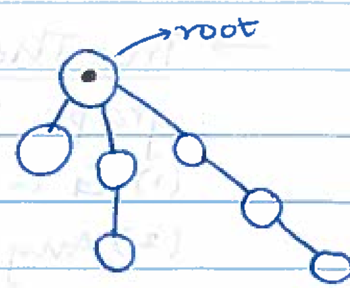
→ Tree : Undirected, acyclic, connected graph

every vertex is reachable from every other





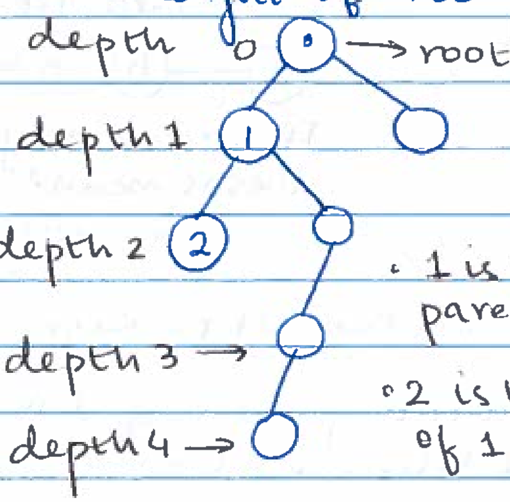
Same graph rooted at different nodes



Outside info → which node is the root

" Tree rooted at ..... "

- Othe terms associated with a tree:
    - root
    - child / parent
    - leaf : vertex without children
    - depth of u :
    - height of tree : largest depth



- 1 is the parent of 2
- 2 is the child of 1

Application:
- Any hierarchical structure.
- File systems

(Q) which node is the root?

(A) It's upto you! or according to definition from outside → "Tree rooted at ....."

→ Tree Theorem: Let $G=(V,E)$ be an undirected graph. Then the following are equivalent:

(1) G is a tree

(2) Any two vertices are connected by a unique simple path

(3) G is connected, but if any edge is removed, it's not. (minimally connected)

(4) G is ~~connected~~ acyclic, but if any edge is added, the graph will contain a cycle.

Notation for size of

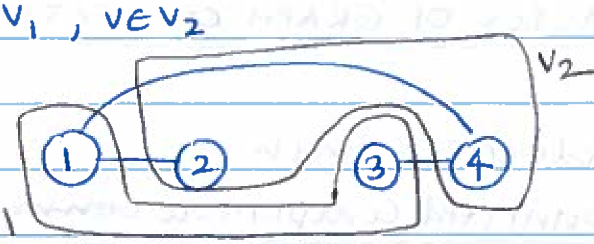
$|E| \rightarrow$ No. of edges
$|V| \rightarrow$ No. of vertices
Size of set of vertices

(5) G is connected and $|E| = |V| - 1$

(6) G is acyclic and $|E| = |V| - 1$

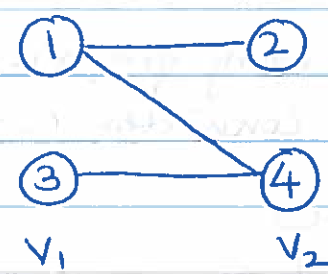
→ Bipartite Graph: Undirected graph in which V can be partitioned into V_1 and V_2 s.t. every edge $\{u,v\}$ has $u \in V_1$, $v \in V_2$

every vertex is in exactly one of V_1 or V_2



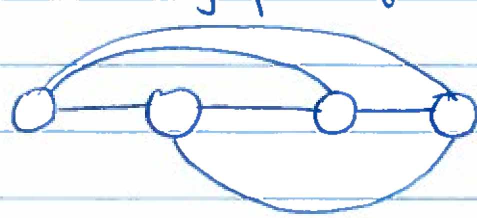
1 & 4, 2 & 3 → a partition? → No; edge between 1 & 4

Can be redrawn as



- No edge within the same partition.
- Application:
  - Connections between routers and computers where only routers and computers can be connected
  - Will be seen later in stable marriage problem.

→ Complete Graph: Undirected graph with an edge between every pair of nodes.



In a sense opposite to minimally connected tree as this has maximum number of edges

? How many edges are there with n vertices?

(A) ${}^nC_2 = \frac{n(n-1)}{2}$ edges for n vertices.

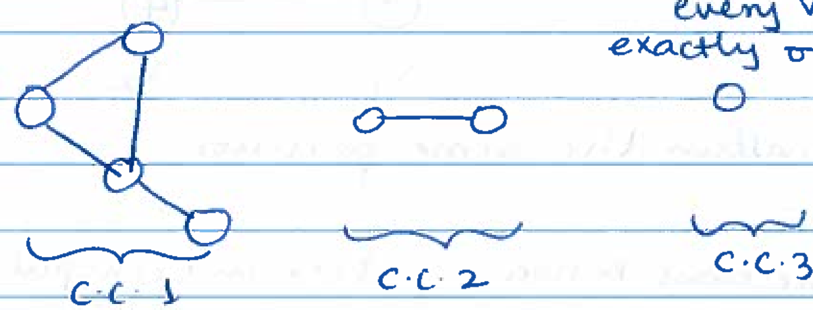
⟹ BASICS OF GRAPH CONNECTIVITY:

1. Undirected Graph:

Connected Component ~~connect~~ containing vertex u: set of all vertices reachable from u

- Note: Every graph can be partitioned into connected components

every vertex belongs to exactly one connected component



- Tree: exactly one connected component.