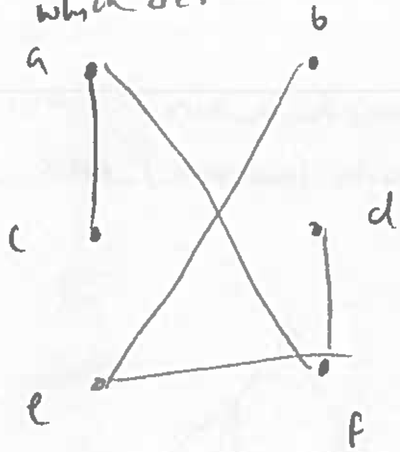
Trees

Now we'll focus on a particular type of graph: Trees.

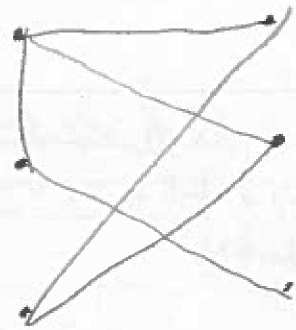
Def: A *tree* is a connected undirected graph with no simple circuits.

Note ⟹ Trees are simple since we cannot have circuits.

Ex: which are trees?



Yes

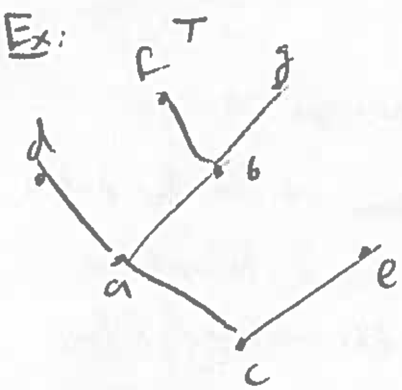


No, a, d, e, b, a is a cycle.

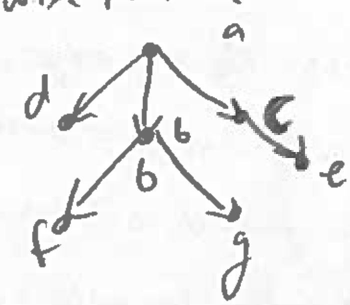
Theorem: An undirected graph is a tree iff there is a unique simple path between any two vertices.

Def: A *rooted tree* is a tree where one vertex is designated the root & every edge is directed away from the root.

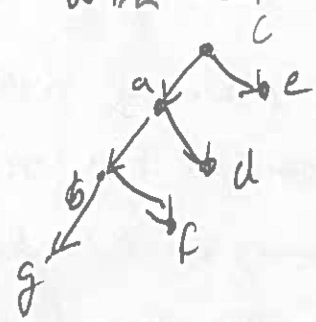
Ex:



with root a



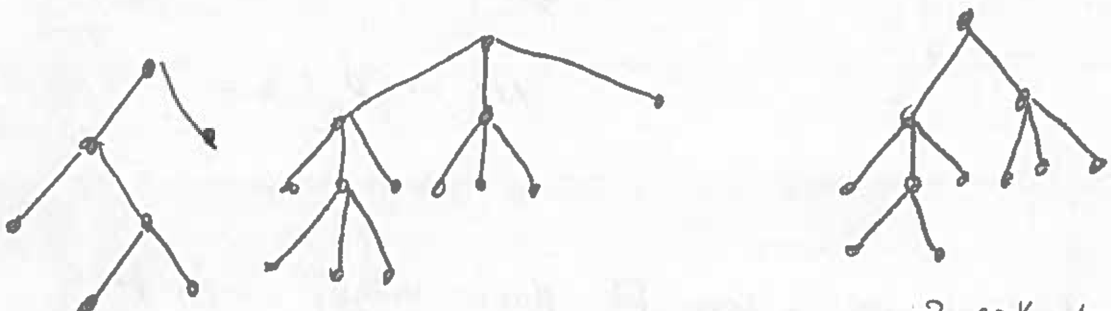
with root c



Def: If T is a rooted tree & v is a vertex. The parent of v is the unique vertex u s.t. ∃ a directed edge u to v. Then v is the child of u. Vertices with the same parent are siblings. The descendants of v are the nodes which have v as an ancestor. A vertex is a leaf if it has no children, a root, if it has no parent. All other nodes are internal vertices. If a is a vertex in a tree the subtree with a as its root is the subgraph consisting of a & all its descendants.

Def: A rooted tree is an m-ary tree if every vertex has no more than m children. It is a full m-ary tree if all internal nodes (and root) have exactly m children.

Ex:



Full binary   full 3-ary tree ternary   3-ary tree. Ternary

Def: An ordered rooted tree is a rooted tree where each child is ordered, usually this is just left vs. right (in binary).

Theorem: A tree with n vertices has n-1 edges.

Pf: By Induction Base case n=1 with 1 vertex there are no edges n-1.

IH: Assume thm holds for a graph of n=k vertices, we show it holds for n=k+1 vertices. This tree of k+1 nodes has atleast one leaf, say v. It only has deg(v)=1 otherwise there is a cycle. Our graph without this node/edge has k nodes and is a tree, by IH it has k-1 edges. Thus adding our node/edge back gives k+1 nodes & k edges.

This closes the induction. □