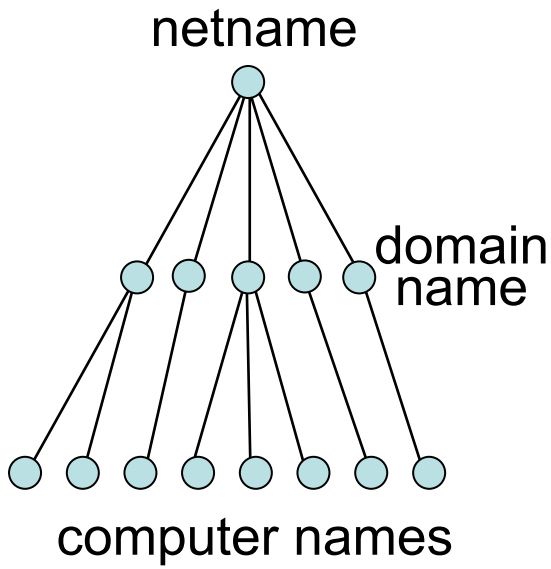
Lesson 13:

Trees

- Basic Terminology
- Graphs vs Trees
- M-Ary trees
- Properties of trees

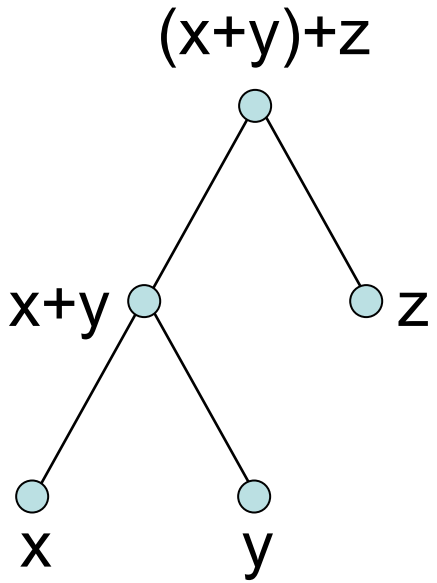


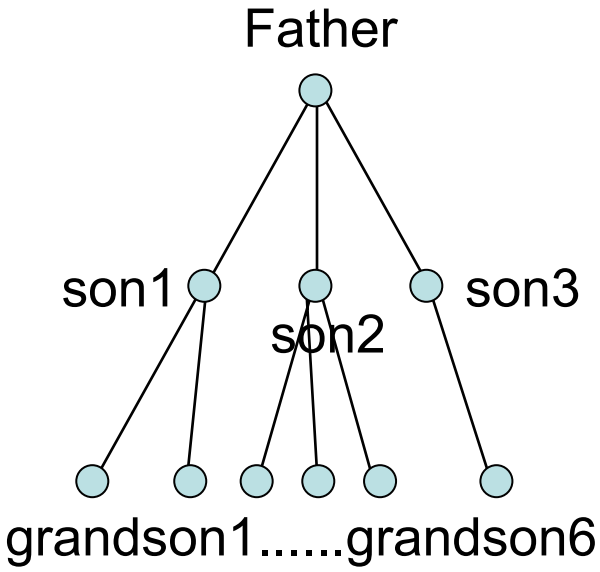
Chapter 9

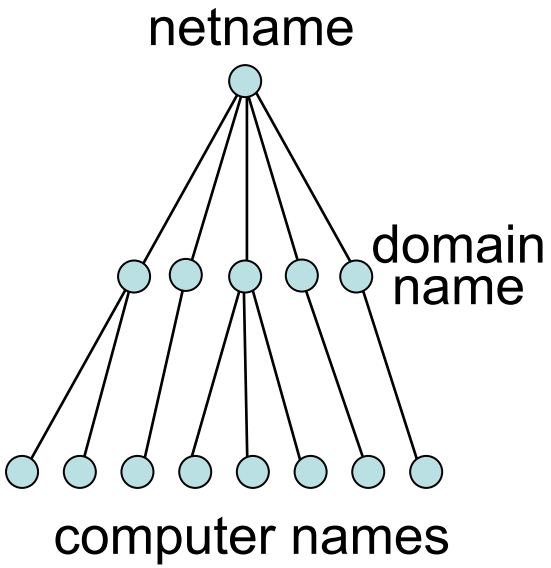
Trees

A Tree is a particular type of graph.

- Computing number of possibilities in counting tasks.
- Developing efficient code (coding, search)
- Study of games and winning strategies.
- Decision trees



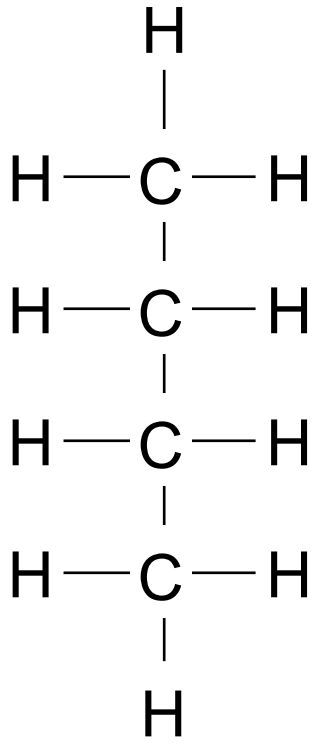


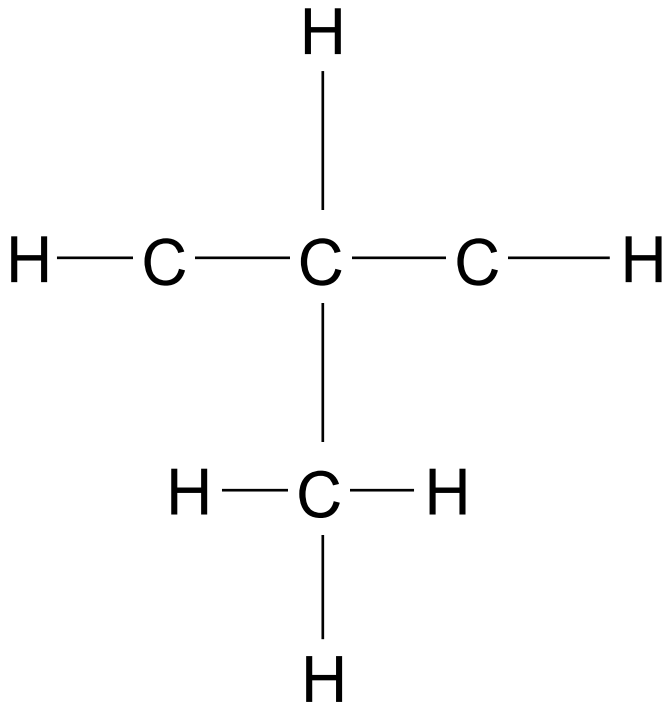


Trees

First use of trees:

1857 - Describing chemical compounds





Trees

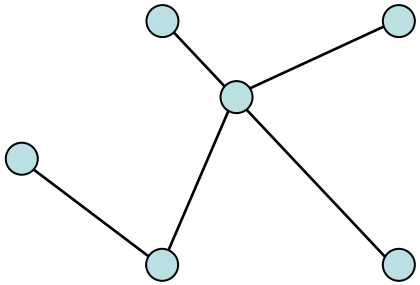
Definition:

An *Tree* (עץ) is a connected undirected graph with no simple circuits.

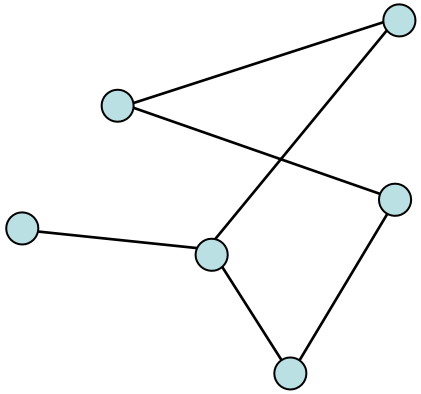
A tree has no circuits, ‘loops’ or parallel edges.

Trees

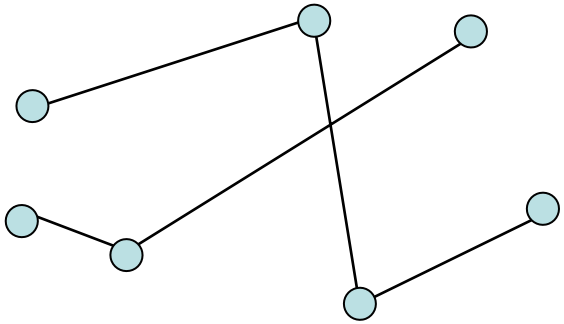
Examples: Trees ?



Yes



No - contains a circuit.

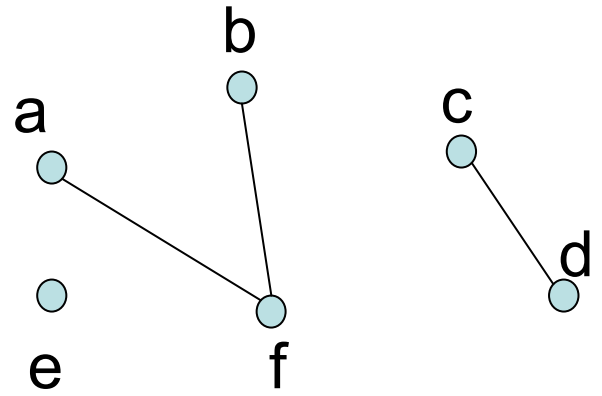


No - is not connected.

Trees

Definition:

A non-connected undirected graph with no simple circuits is called a *Forest* (יער) .



Trees

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

Proof:

1) T is a tree $\rightarrow$ there is a unique simple path.
   Let x,y be vertices. T is connected so there is a simple path between them. If it is not unique then the 2 paths can be combined into a circuit - contradicting that T is a tree.

Trees

Proof cont:

2) Unique simple paths $\rightarrow$ T is a tree.

T is connected - because every 2 vertices are connected.

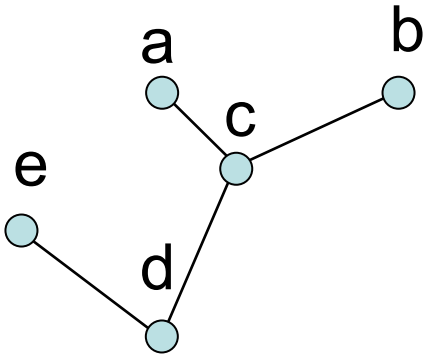
T has no circuits, because if it had then we could divide the circuit into 2 paths connecting a pair of vertices - contradiction.

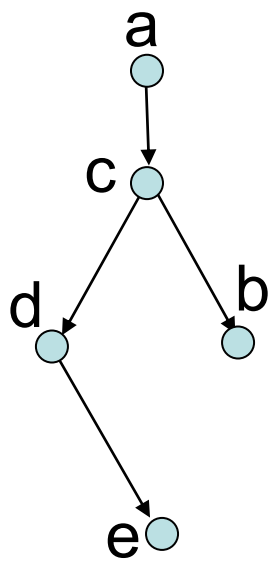
Trees

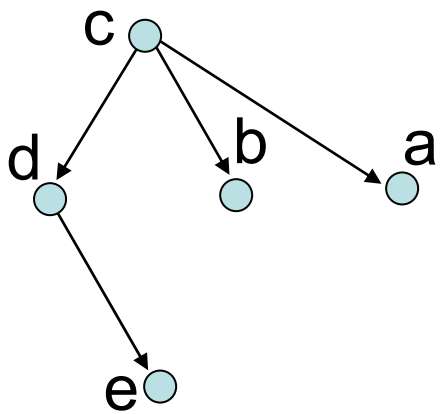
Some applications designate one of the vertices as special and called the *Root* (שורש).

A rooted tree can change an undirected tree to a directed tree.

Every edge is directed away from the root.







Trees - Terminology

Root (שורש) - the (only) vertex with in-degree = 0.

Parent (אב/הורה) of a vertex v is the (unique) vertex u for which there is an edge (u,v).

Child (בן/ילד) of vertex v is a vertex u for which v is its parent.

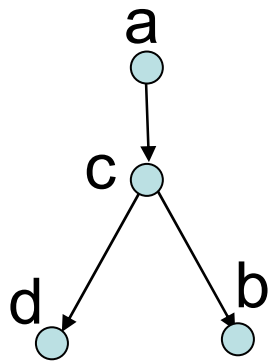
Sibling (אחים) - 2 vertices with the same parent.

Descendents (צאצאים) of a vertex v are all the nodes that have a directed path from v (i.e. all its children, children of children etc...).

Leaf (עלה) - a vertex with no child.

Internal Vertex/node - a vertex that is not a leaf.

Subtree (תת-עץ) with root u is the subgraph containing u and all its descendents.



M-Ary Trees

Definition:

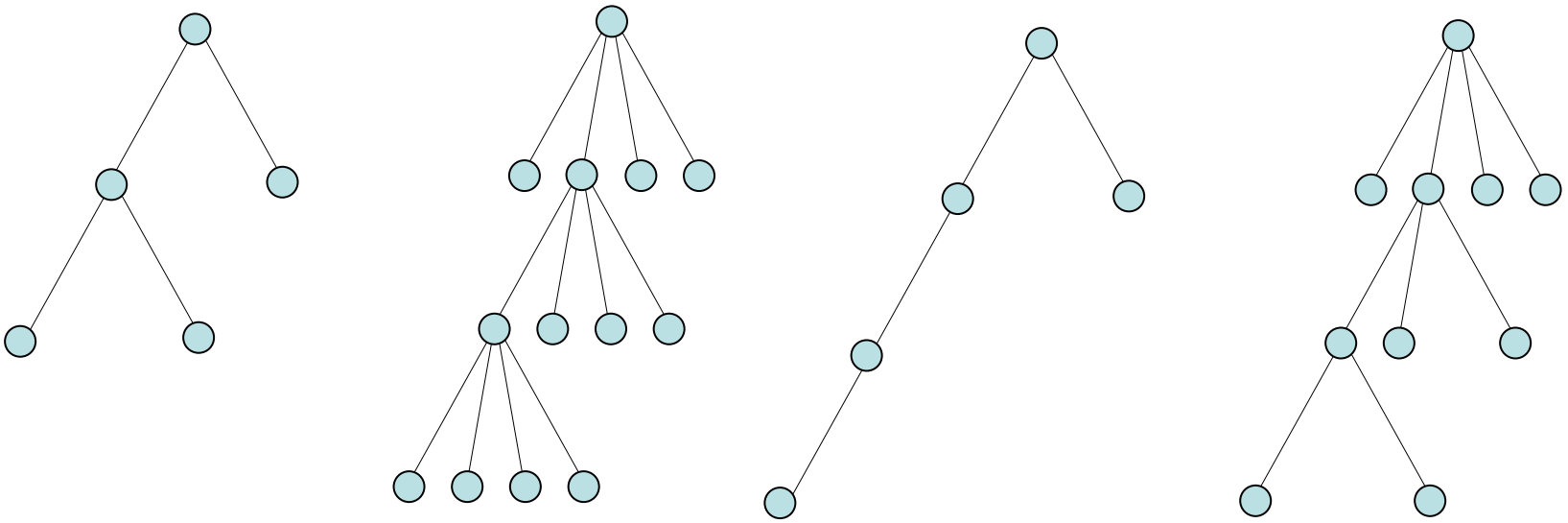
An *m-ary tree* (עץ m-י) is a tree for which every internal node has at most m children.

A *full m-ary tree* (עץ m-י מלא) is a tree for which every internal node has exactly m children.

A Binary tree (עץ בינארי) is a tree with at most 2 children at each internal node.

M-Ary Trees

Examples:



full binary tree

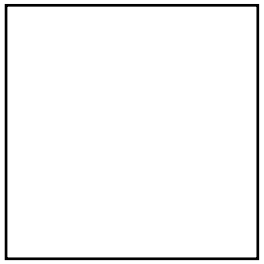
full 4-ary tree (Quadtree)

not full binary tree

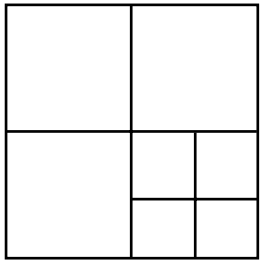
not full 4-ary tree (Quadtree)

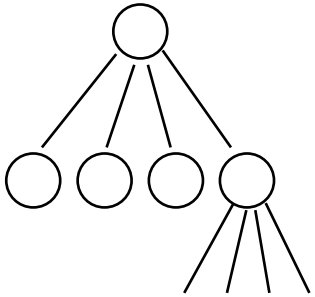
4-ary Trees (Quadtrees)

Image

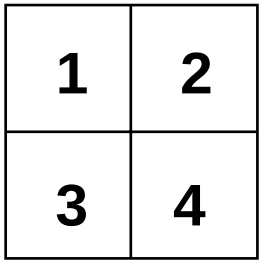


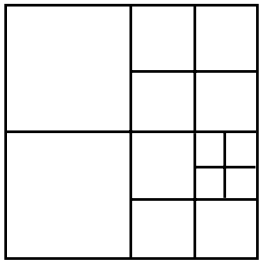
Quad Tree



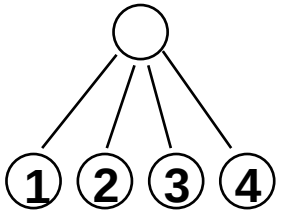


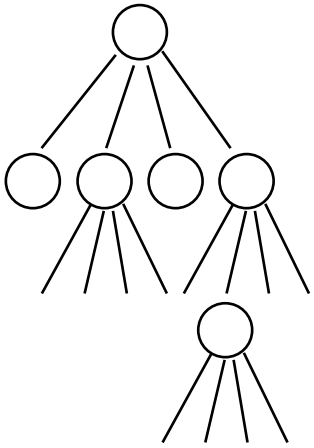
Image





Quad Tree





Properties of Trees

Theorem:

In a tree with n vertices there are exactly n-1 edges.

Proof of Theorem: Using 2 lemmas.

Lemma 1:

If a graph with n vertices has n edges or more then the graph has a circuit.

Lemma 2:

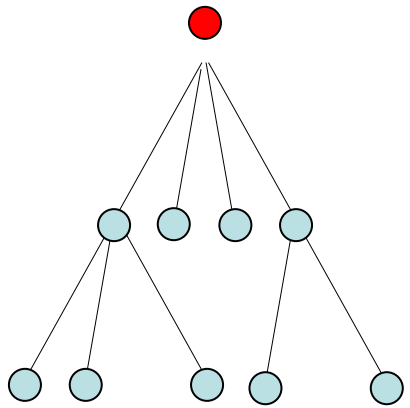
If a graph with n vertices has less than n-1 edges then it is not connected.

Proof of Lemmas: By induction on n.

Properties of Trees

Aternative Proof of Theorem: Combinatorial proof.

In a tree with n vertices, choose a vertex r as the root.
There is a 1:1: and onto mapping between every edge and its destination vertex.
There are n-1 vertices (without the root) so there are n-1 edges.



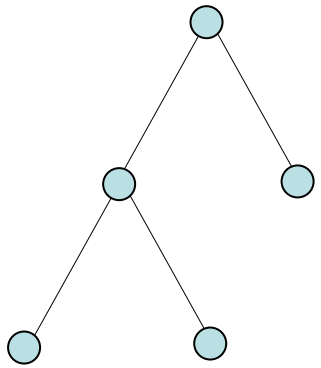
Properties of Trees

Theorem:

A full m-ary tree with k internal nodes has a total of m*k + 1 vertices.

Proof :

Every node except the root is a child of an internal node. Every internal node has m children, thus there are k*m nodes except the root and a total of k*m+1 nodes in tree.



Properties of Trees

Question:

How many leaves in a full m-ary tree with k internal nodes?

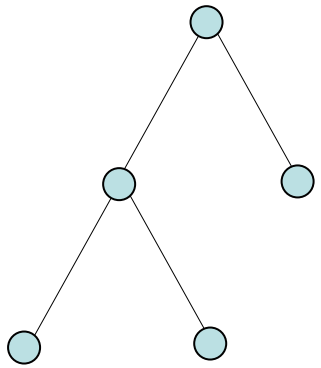
Answer:

There are m*k+1 vertices in total,

k are internal thus:

there are

l = m*k+1 - k = (m-1)*k + 1 leaves.



Properties of Trees

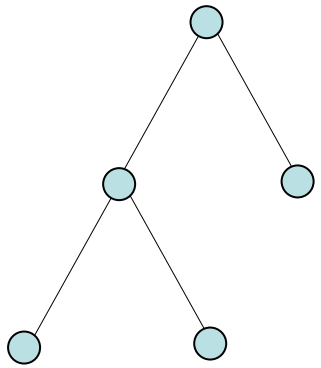
Question:

How many internal nodes in a full m-ary tree with l leaves?

Answer:

There are l = (m-1)*k + 1 leaves

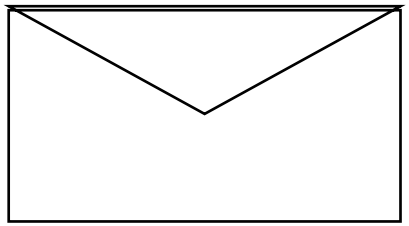
thus k = (l-1)/(m-1) internal nodes.



Trees

Question:

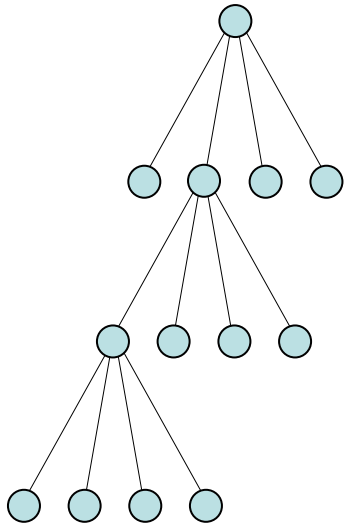
A chain letter is sent. Each person is asked to send the letter to 4 other people. Some do and some don't send at all. How many people have read the letter (including the 1st) if no one receives twice and the chain stops after there are 100 people who have read the letter but did not respond.



Trees

Answer:

The chain letter is represented as a full 4-ary tree:



Number of leaves l = 100.

k = (l-1)/(m-1) = 99/3 = 33 internal nodes

Total = l+k = 100 + 33 = 133.

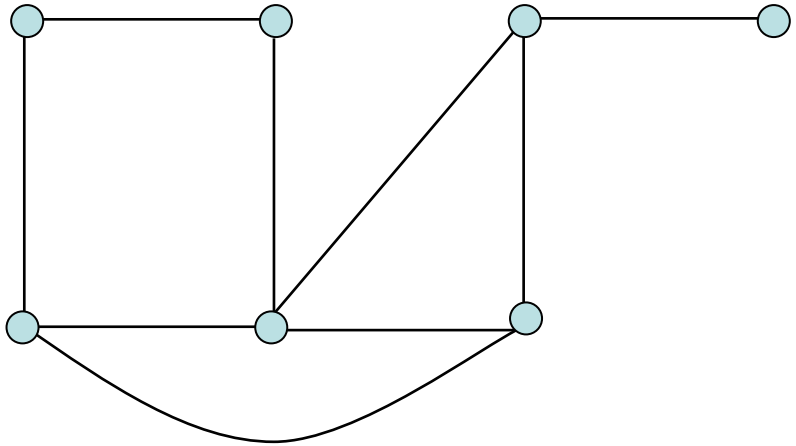
Spanning Trees

Definition:

A *Spanning Tree* (עץ פורש) of a simple graph G is a subgraph of G that is a tree and contains every vertex of G.

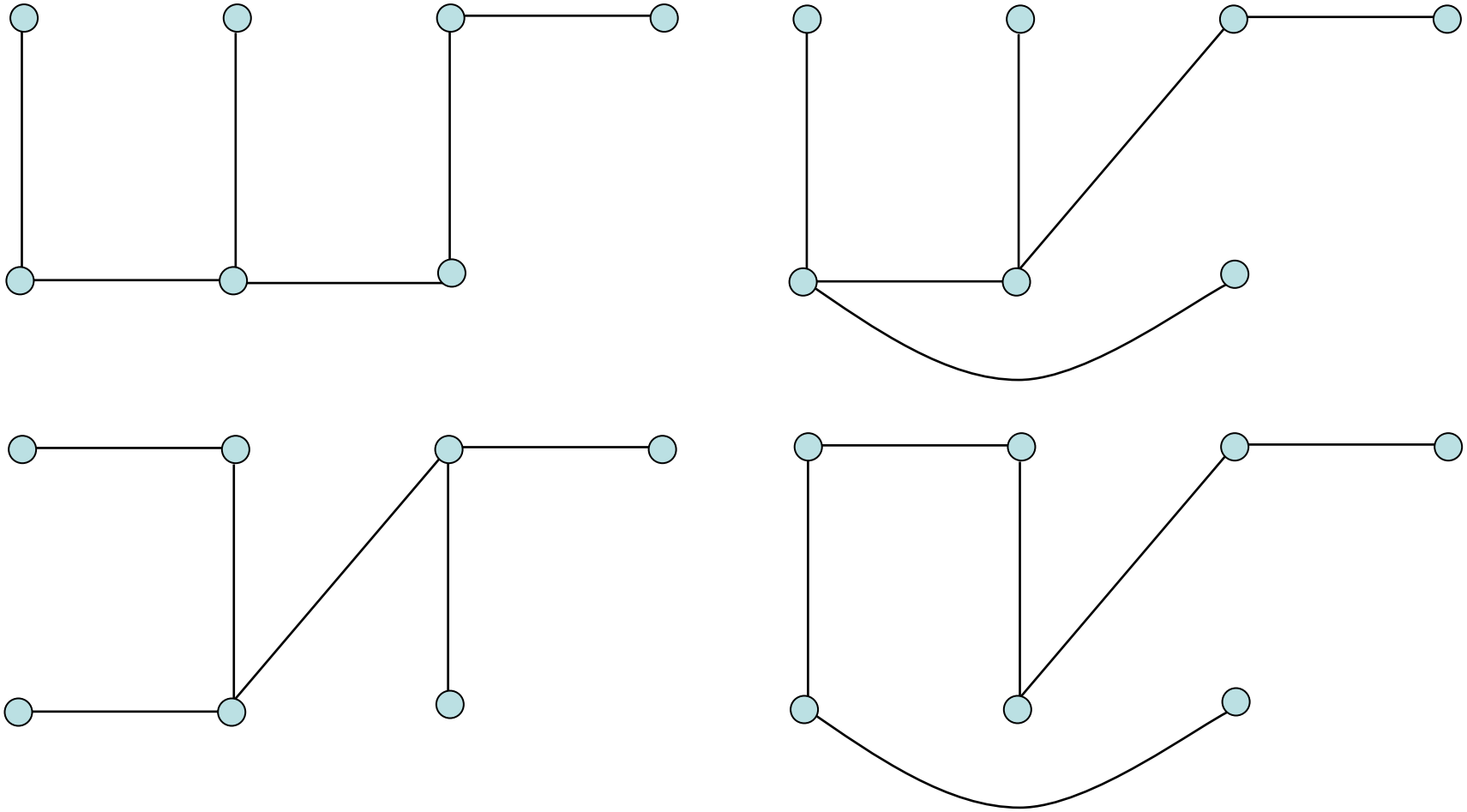
Spanning Trees

Example:



Spanning Trees

Example:



Spanning Trees

Theorem:

A simple graph is connected iff it has a spanning tree.

Proof :

If T is a spanning tree of G then there is a path between every 2 vertices of T, however every vertex of G is in T so every 2 vertices of G have a path so G is connected.

Spanning Trees

Proof cont:

If G is connected and is not already a tree, then it contains a circuit.

Remove one of its edges creating a subgraph of G.

Every 2 vertices still have a path (if the path contained the deleted edge, then replace the edge in the path by the rest of the circuit).

Continue removing edges in this manner maintaining connectivity of the subgraph until no circuits left.

Process ends because number of edges is finite.

Trees - Fun

Trees - Fun

So many trees - one does not see the forest.