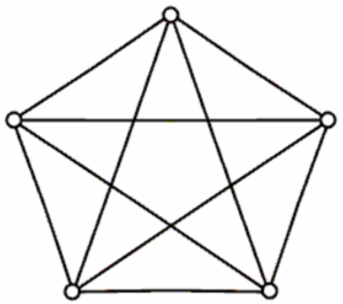
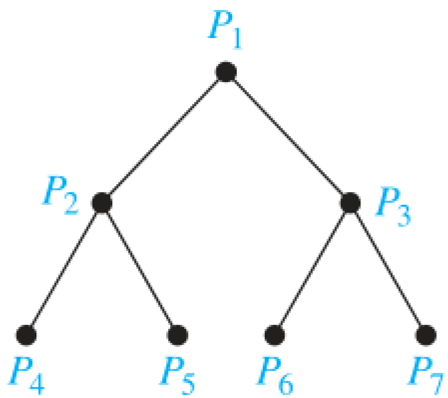
Trees

Chapter 11

- Trees are used as models in such diverse area as computer science, chemistry, geology, botany.
- trees are used to construct efficient algorithms for locating items in a list.
- Trees can be used to construct efficient codes saving costs in data transmission and storage.
- a tree-connected network can be used for parallel computation
- Using weighted trees to construct networks containing the least expensive set of lines linking different network nodes.



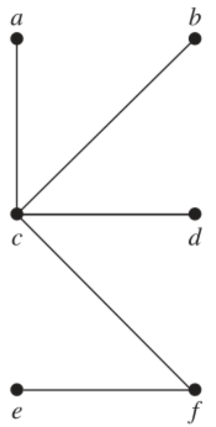
Communication network connecting its five Computer centers



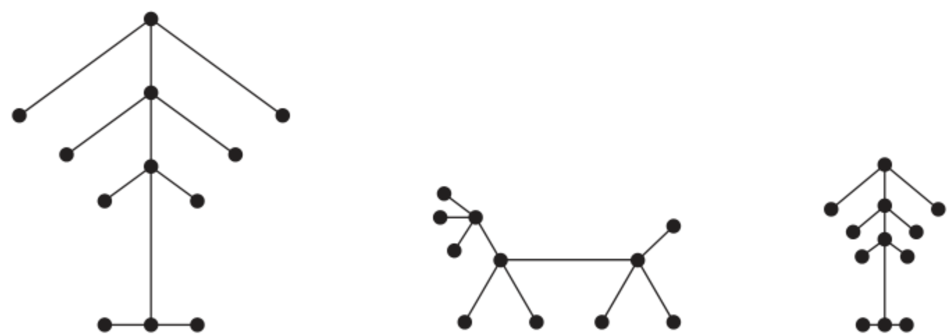
Tree connected network of seven processors

Undirected graph

- **Definition:** A tree is a connected undirected graph with no simple circuits.
- Leaf: the degree of vertex is 1.
- Internal vertex: the degree of vertex is at least 2.
- The graphs that are not connected and contains no simple circuits are called **forests** iff each of their connected components is a tree.



This is one graph with three connected components.



Theorem

- An undirected graph is a tree iff there is a unique simple path between any two of its vertices.
- A tree with n vertices has n-1 edges.
- If one of these edges is deleted in the tree, this tree become unconnected.
- If add one edge to this tree, this graph has circuit.

Directed graph

- In many applications of trees a particular vetex is designated as the **root**.
- A tree together with its root produces a directed graph called a **rooted tree**.
- We can change an unrooted tree into a rooted tree by choosing any vertex as the root. Different choices of root produce different rooted trees.

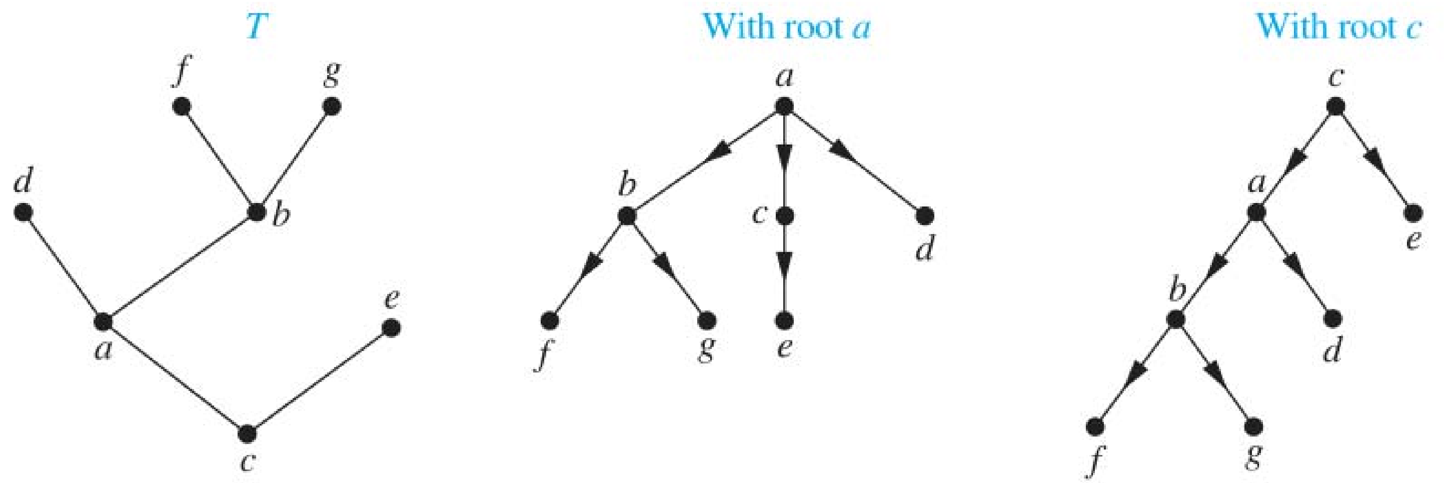
definitions

Root tree: The in-degree of one vertex is 0, the in-degrees of other vertices are all 1.

Root： the in-degree is 0

height： the maximum of the levels of vertices.

Level： the length of the unique path from the root to this vertex. The level of the root is defined to be zero

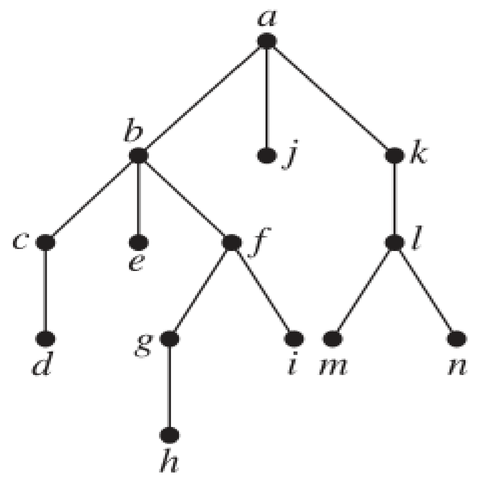


The terminology for tree comes from botany and genealogy.

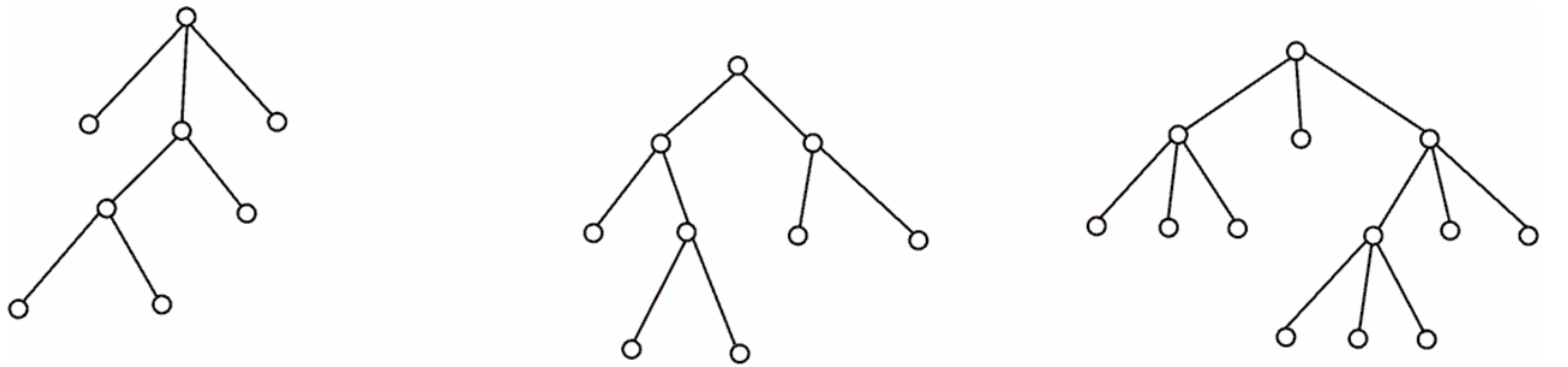
Suppose T is a rooted tree. If u and v are vertices in T other than root,

- The **parent** of *v* is the unique vertex *u* such that there is a directed edge from *u* to *v*.
- When *u* is the parent of *v*, *v* is called a **child** of *u*.
- Vertices with same parent are called **siblings.**
- The **ancestors** of a vertex other than the root are the vertices in the path from the root to this vertex, excluding the vertex itself and including the root.

- The **descendants** of a vertex *v* are those vertices that have *v* as an ancestor.
- A vertex of a tree is called a **leaf** if it has no children.
- Vertices that have children are called **internal vertices**.
- If *a* is a vertex in a tree, the **subtree** with *a* as its root is the subgraph of the tree consisting of *a* and its descendants and all edges incident to these descendants.



Definition : A rooted tree is called an ***m-ary*** tree if every internal vertex has no more than *m* children. The tree is called a ***full m-ary tree*** if every internal vertex has exactly *m* children. An m-ary tree with m=2 is called a ***binary tree***.



- An **ordered rooted tree** is a rooted tree where the children of each internal vertex are ordered.
- Ordered rooted tree are drawn so that the children of each internal vertex are shown in order from left to right.
- In an ordered binary tree, if an internal vertex has two children, the first child is called the **left child** and the second child is called the **right child**.
- The tree rooted at the left child of a vertex is called the **left subtree** of this vertex, the tree rooted at the right child of a vertex is called the **right subtree** of this vertex.

THEOREM

A full m-ary tree with i internal vertices contains $n=mi+1$ vertices.

Proof: Every vertex, except the root, is the child of an internal vertex. Since each of the i internal vertices has m children, there are mi vertices in the tree other than root. Therefore, the tree contains $n=mi+1$ vertices.

THEOREM

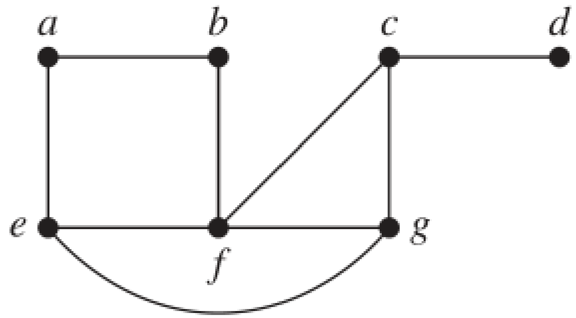
Suppose that T is a full *m*-ary tree. Let *i* be the number of internal vertices and *l* the number of leaves in this tree.

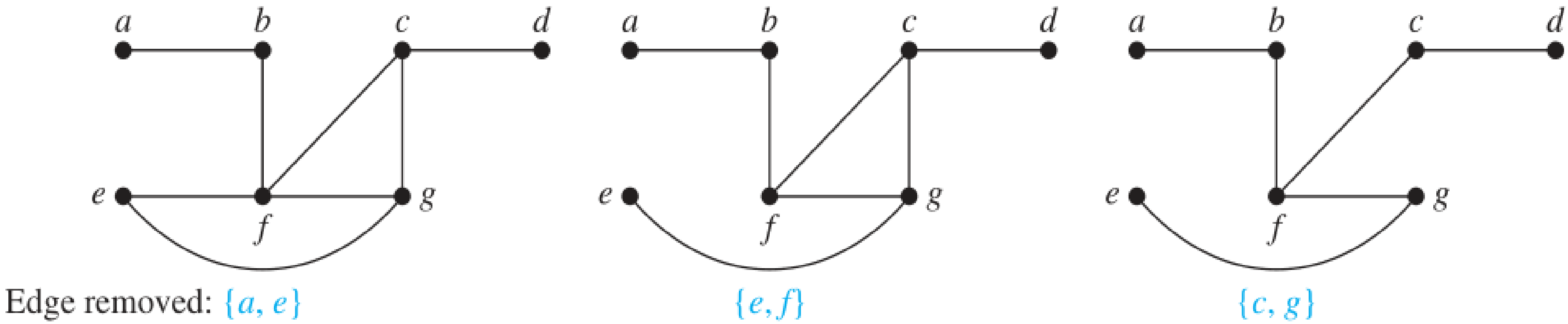
A full m-ary tree with

1. *n* vedrtices has $i=(n-1)/m$ internal vertices and $l=[(m-1)n+1]/m$ leaves;
2. *i* internal vertices has $n=mi+1$ vertices and $l=(m-1)i+1$ leaves;
3. *l* leaves has $n=(ml-1)/(m-1)$ vertices and $i=(l-1)/(m-1)$ internal vertices.

Spanning tree

Definition: Let G be a simple graph. A spanning tree of G is a subgraph of G that is a tree containing every vertex of G.





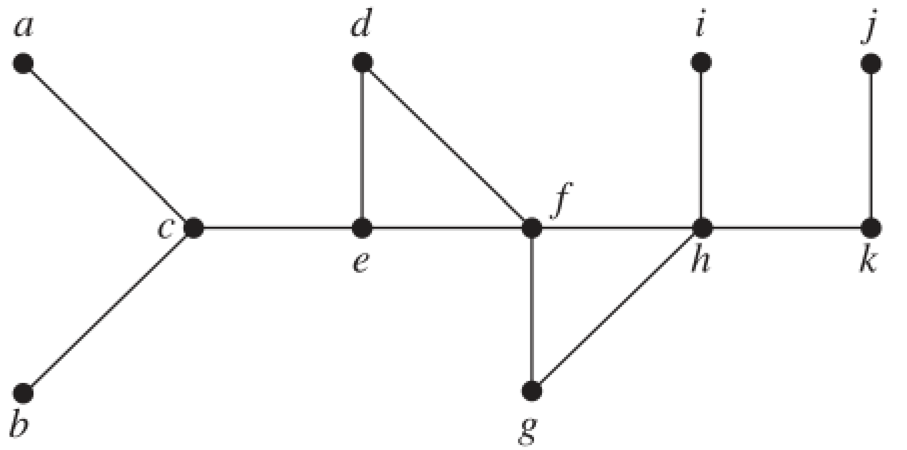
Theorem

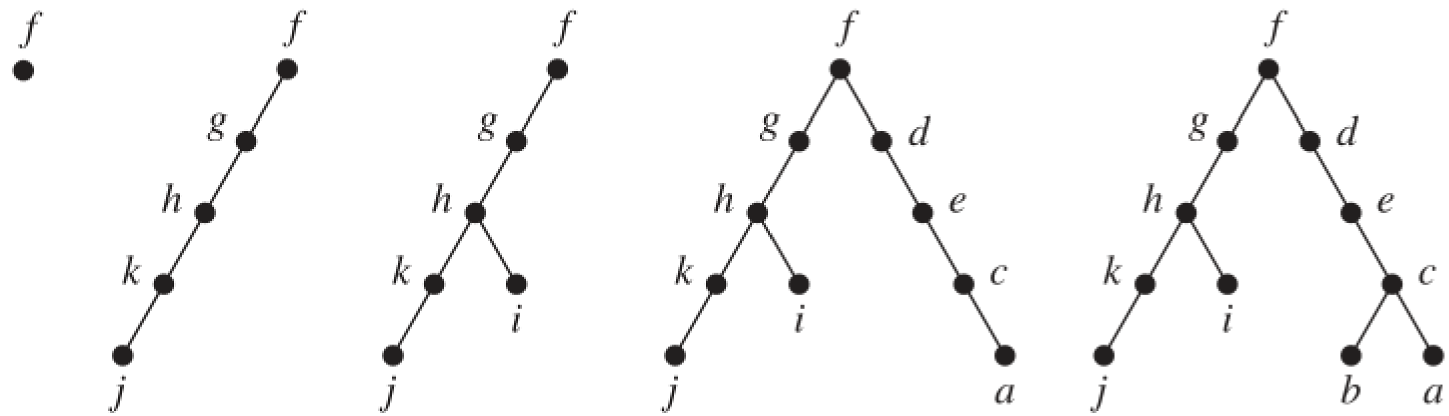
1. A simple graph is connected if and only if it has a spanning tree
2. G is connected graph with n vertices and m edges, then $m \geq n-1$.

Depth-First Search

- Arbitrarily choose a vertex of the graph as the root.
- starting at this vertex, successively adding vertices and edges to form path as long as possible.
- if the path does not go through all vertices, move back to the next to last vertex in the path, and, if possible, form a new path, starting at this vertex passing through vertices that were not already visited.
- Repeat this procedure, beginning at the last vertex visited, moving back up the path one vertex at a time, forming new paths that are as long as possible until no more edges can be added.

Example





(a) (b) (c) (d) (e)

minimum spanning tree

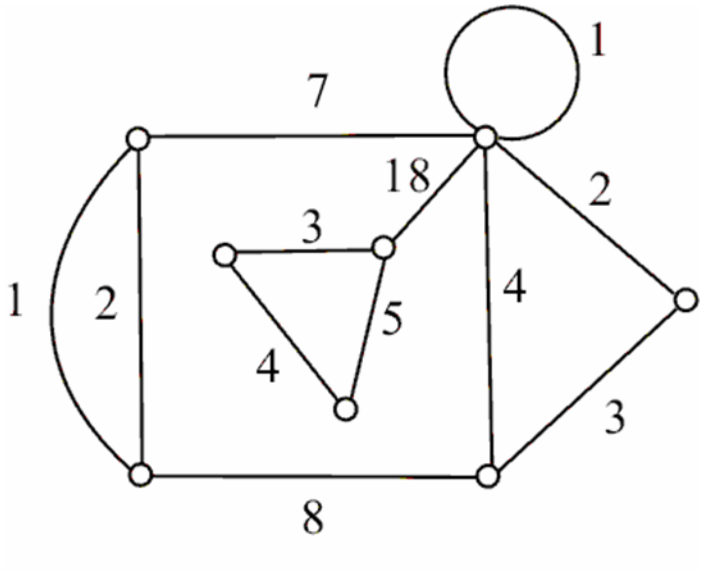
A minimum spanning tree in a connected weighted graph is a spanning tree that has the smallest possible sum of weights of its edges.

Kruskal's algorithm

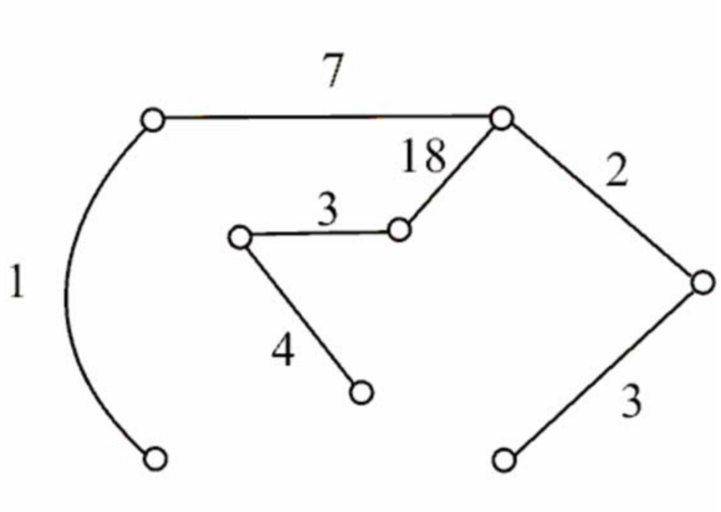
G: weighted connected undirected graph with n vertices

1. $E=\{\}$
2. 2. Select the edge with the least weight, and not making a cycle with members of E
3. Repeat until E contains n-1 edges.

Example



Undirected graph



minimum spanning tree

Homework

- 11.1 1,3
- 11.4 1,2, 3,
- 11.5 5,7,