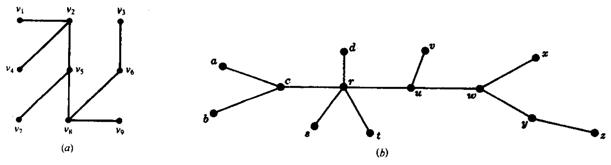
**Spanning Tree and Binary Tree Traversal**

**Tree Graphs**

A graph T is called a tree if T is connected and T has no cycles. Examples of trees are shown in Figure. A forest G is a graph with no cycles; hence the connected components of a forest G are trees. [A graph without cycles is said to be cycle-free.] The tree consisting of a single vertex with no edges is called generate tree.



Consider a tree T. Clearly, there is only one simple path between two vertices of T, otherwise, the two paths would form a cycle. Also:

Suppose there is no edge {u,v} in T and we add the edge e = {u,v} to T. Then the simple path from u to v in T and e will form a cycle, hence T is no longer a tree.

On the other hand, suppose there is an edge e = {u,v} in T, and we delete e from T. Then T is no longer connected (since there cannot be a path from u to v); hence T is no longer a tree.

The following theorem (proved in Problem 1-16) applies when our graphs are finite.

**Theorem**

Let G be a graph with n > 1 vertices. Then the following are equivalent:

(i) G is a tree.

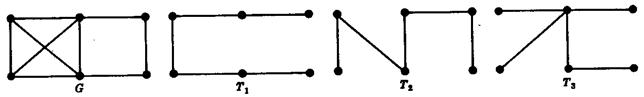
(ii) G is a cycle-free and has n - 1edges.

(iii) G is connected and has n - 1 edges.

This theorem also tells us that a finite tree T with n vertices must have n – 1 edges. For tree in Fig.(a) has 9 vertices and 8 edges, and the tree in Fig.(b) has 13 vertices and 12 edges.

**Spanning Trees**

A subgraph T of a connected graph G is called a spanning tree of G if T is a tree and includes all the vertices of G. Following figure shows a connected graph G and spanning trees T1, T2, and T3 of G.



**Minimum Spanning Trees**

Suppose G is a connected weighted graph. That is, each edge of G is assigned a nonnegative number called the weight of the edge. Then any spanning tree T of G is assigned a total weight adding the weights of the edges in T. A minimal spanning tree of (T is a spanning tree whose is as small as possible.

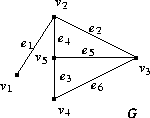
**Examples-1**

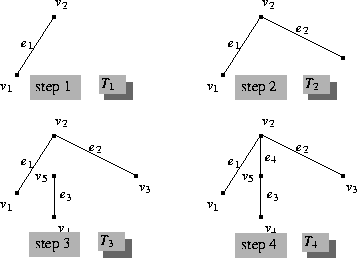
Graph *G*

|  |  |  |
| --- | --- | --- |
| http://staff.scm.uws.edu.au/~zhuhan/dm_notes/q/q1.gif  Removing edge *{v2,v3}* from *G* to break the cycle *v5 http://staff.scm.uws.edu.au/~zhuhan/dm_notes/tex/rightarrow.gif v2 http://staff.scm.uws.edu.au/~zhuhan/dm_notes/tex/rightarrow.gif v3 http://staff.scm.uws.edu.au/~zhuhan/dm_notes/tex/rightarrow.gif v5* | http://staff.scm.uws.edu.au/~zhuhan/dm_notes/q/q2.gif  Removing edge *{ v4,v3}* to break the last remaining cycle *v5 http://staff.scm.uws.edu.au/~zhuhan/dm_notes/tex/rightarrow.gif v3 http://staff.scm.uws.edu.au/~zhuhan/dm_notes/tex/rightarrow.gif v4 http://staff.scm.uws.edu.au/~zhuhan/dm_notes/tex/rightarrow.gif v5*. | http://staff.scm.uws.edu.au/~zhuhan/dm_notes/q/q3.gif  Spanning tree |

Of course, there can be different spanning trees for the same connected graph *G*.

However, we can also construct a spanning tree in the opposite way, that is, by gradually collecting edges from *G*, and making sure that the resulting subgraph is always connected and contains no cycles. For instance, a spanning tree *T4*, different from that obtained in example-1, can derived from the following 4 steps





**Example-2**

|  |  |
| --- | --- |
| http://staff.scm.uws.edu.au/~zhuhan/dm_notes/q/q4.gif | http://staff.scm.uws.edu.au/~zhuhan/dm_notes/q/q5.gif |

**No. of Spanning Trees**

|  |  |
| --- | --- |
| **Kirchoff’s theorem** is useful in finding the number of spanning trees that can be formed from a connected graph. The diagonal values are replaced with the degree of vertices and all other elements with – 1, then separate the cofactor and the computational result is called the No. of Spanning Trees.  The matrix ‘A’ be filled as, if there is an edge between two vertices, then it should be given as ‘1’, else ‘0’. |  |

