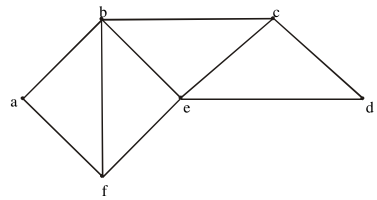
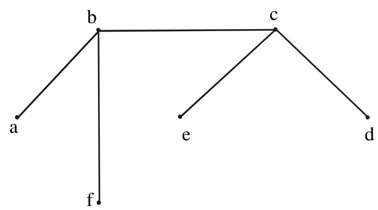
SPANNING TREES:

Suppose it is required to develop a system of roads between six major cities.
A survey of the area revealed that only the roads shown in the graph could be constructed.



For economic reasons, it is desired to construct the least possible number of roads to connect the six cities.
One such set of roads is



Note that the subgraph representing these roads is a tree, it is connected & circuit-free (six vertices and five edges)

SPANNING TREE:

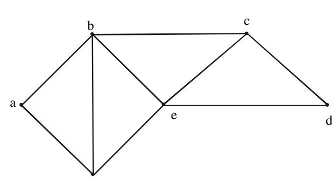
A spanning tree for a graph G is a subgraph of G that contains every vertex of G and is a tree.

REMARK:

1. Every connected graph has a spanning tree.
2. A graph may have more than one spanning trees.
3. Any two spanning trees for a graph have the same number of edges.
4. If a graph is a tree, then its only spanning tree is itself.

EXERCISE:

Find a spanning tree for the graph below:



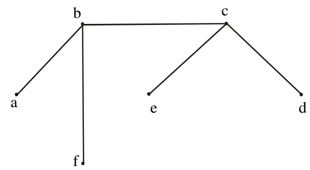
SOLUTION:

The graph has 6 vertices (a, b, c, d, e, f) & 9 edges so we must delete 9 - 6 + 1 = 4 edges (as we have studied in lecture 44 that a tree of vertices n has n-1 edges). We delete an edge in each cycle.

1. Delete af   2. Delete fe
3. Delete be   4. Delete ed

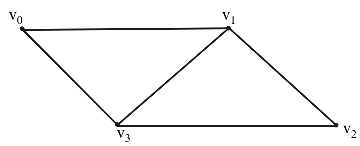
Note it that we can construct road from vertex a to b, but can't go from "a to e", also from "a to d" and from "a to c ", because there is no path available.

The associated spanning tree is



EXERCISE:

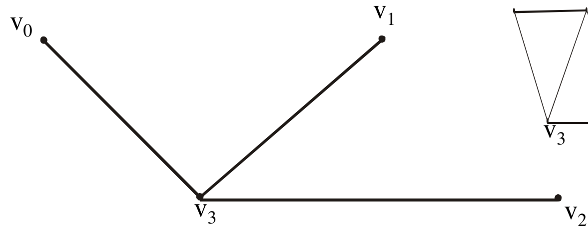
Find all the spanning trees of the graph given below.

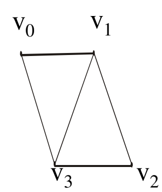


SOLUTION:

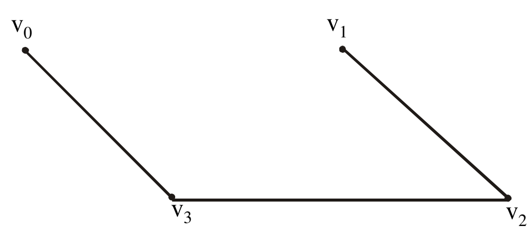
The graph has n = 4 vertices and e = 5 edges. So we must delete e - v + 1 = 5 - 4 + 1 = 2 edges from the cycles in the graph to obtain a spanning tree.

(1) Delete v_0v_1 & v_1v_2 to get

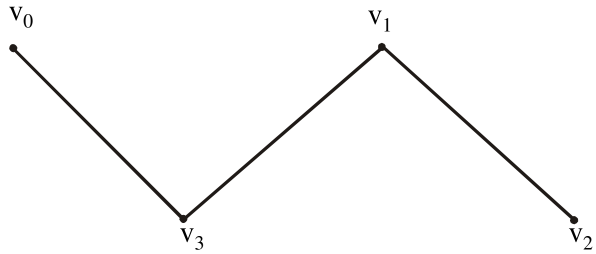


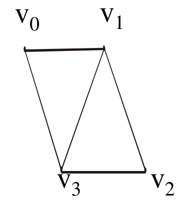


(2) Delete v_0v_1 & v_1v_3 to get

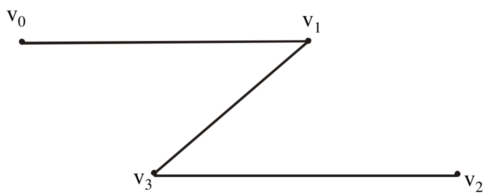


(3) Delete v_0v_1 & v_2v_3 to get

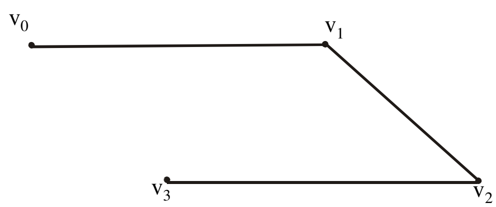


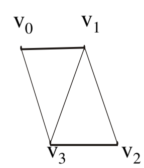


(4) Delete v_0v_3 & v_1v_2 to get

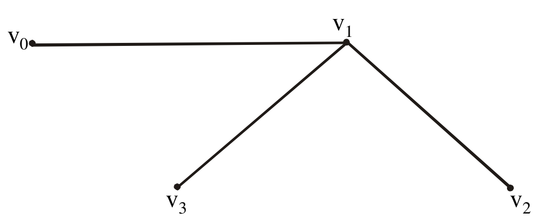


(5) Delete v_0v_3 & v_1v_3 to get

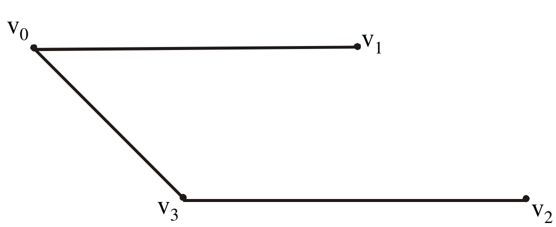


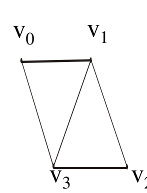


(6) Delete v_0v_3 & v_2v_3 to get

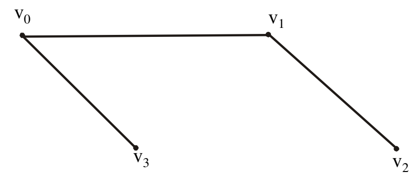


(7) Delete v_1v_3 & v_1v_2 to get





(8) Delete v_1v_3 & v_2v_3 to get



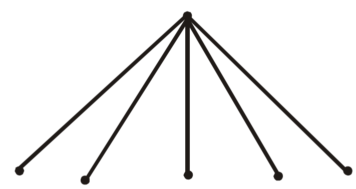
EXERCISE:

Find a spanning tree for each of the following graphs.

(a) $k_{1,5}$ (b) k_4

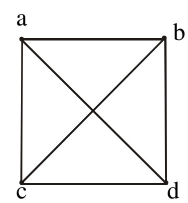
SOLUTION:

(a). $k_{1,5}$ represents a complete bipartite graph on (1,5) vertices, drawn below:



Clearly the graph itself is a tree (six vertices and five edges). Hence the graph is itself a spanning tree.

(b) k_4 represents a complete graph on four vertices.



Now

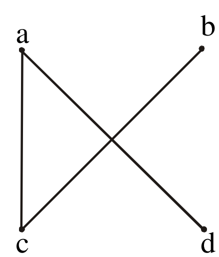
number of vertices = n = 4 and number of edges = e = 6

Hence we must remove

$$e - v + 1 = 6 - 4 + 1 = 3$$

edges to obtain a spanning tree.

Let ab, bd & cd edges are removed. The associated spanning tree is

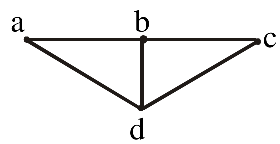


KIRCHHOFF'S THEOREM
OR MATRIX - TREE THEOREM

Let M be the matrix obtained from the adjacency matrix of a connected graph G by changing all 1's to -1's and replacing each diagonal 0 by the degree of the corresponding vertex. Then the number of spanning trees of G is equal to the value of any cofactor of M.

EXAMPLE:

Find the number of spanning trees of the graph G.



SOLUTION:

The adjacency matrix of G is

$$A(G) = \begin{array}{c} \\ a \\ b \\ c \\ d \end{array} \begin{array}{c} \begin{array}{cccc} a & b & c & d \end{array} \\ \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix} \end{array}$$

The matrix specified in Kirchhoff's theorem is

$$M = \begin{bmatrix} 2 & -1 & 0 & -1 \\ -1 & 3 & -1 & -1 \\ 0 & -1 & 2 & -1 \\ -1 & -1 & -1 & 3 \end{bmatrix}$$

Now cofactor of the element at (1,1) in M is

$$\begin{vmatrix} 3 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 3 \end{vmatrix}$$

Expanding by first row, we get

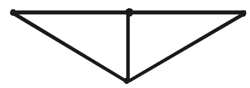
$$= 3\begin{vmatrix} 2 & -1 \\ -1 & 3 \end{vmatrix} - (-1)\begin{vmatrix} -1 & -1 \\ -1 & 3 \end{vmatrix} + (-1)\begin{vmatrix} -1 & 2 \\ -1 & -1 \end{vmatrix}$$

$$= 3(6-1) + (-3-1) + (-1)(1+2)$$

$$= 15 - 4 - 3 = 8$$

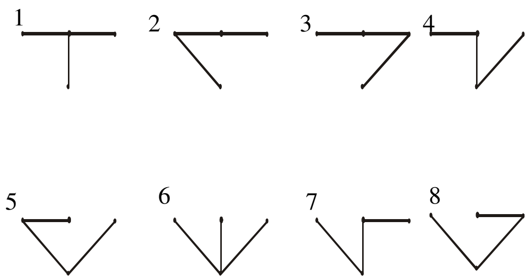
EXERCISE:

How many non-isomorphic spanning trees does the following simple graph has?



SOLUTION:

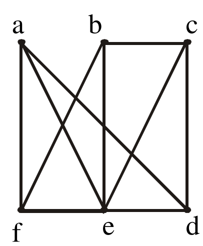
There are eight spanning tree of the graph



Clearly 1 & 6 are isomorphic, and 2, 3, 4, 5, 7, 8 are isomorphic. Hence there are only two non-isomorphic spanning trees of the given graph.

EXERCISE:

Suppose an oil company wants to build a series of pipelines between six storage facilities in order to be able to move oil from one storage facility to any of the other five. For environmental reasons it is not possible to build a pipeline between some pairs of storage facilities. The possible pipelines that can be build are.

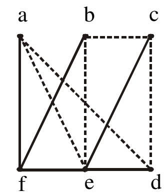


Because the construction of a pipeline is very expensive, construct as few pipelines as possible.

(The company does not mind if oil has to be routed through one or more intermediate facilities)

SOLUTION:

The task is to find a set of edges which together with the incident vertices from a connected graph containing all the vertices and having no cycles. This will allow oil to go from any storage facility to any other without unnecessary building costs. Thus, a tree containing all the vertices of the graph is to be soughed. One selection of edges is



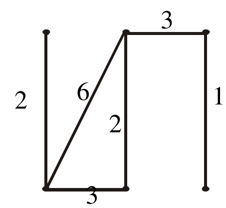
DEFINITION:

A weighted graph is a graph for which each edge has an associated real number weight.

The sum of the weights of all the edges is the total weight of the graph.

EXAMPLE:

The figure shows a weighted graph



with total weight is $2 + 6 + 3 + 2 + 3 + 1 = 17$

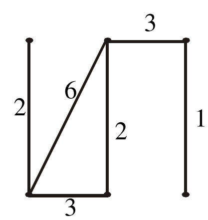
MINIMAL SPANNING TREE:

A minimal spanning tree for a weighted graph is a spanning tree that has the least possible total weight compared to all other spanning trees of the graph.

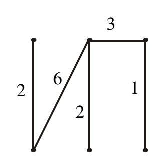
If G is a weighted graph and e is an edge of G then w(e) denotes the weight of e and w(G) denotes the total weight of G.

EXERCISE:

Find the three spanning trees of the weighted graph below. Also indicate the minimal spanning tree.

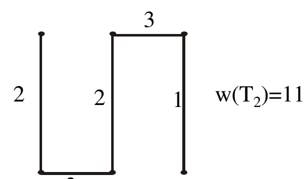


SOLUTION:

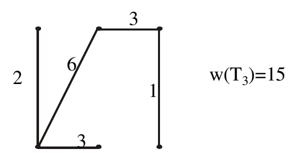


T_1

$w(T_1)=14$



T_2



T_3

T_2 is the minimal spanning tree, since it has the minimum weight among the spanning trees.

KRUSKAL'S ALGORITHM:

Input: G [a weighted graph with n vertices]

Algorithm:

1. Initialize T(the minimal spanning tree of G) to have all the vertices of G and no edges.
2. Let E be the set of all edges of G and let m: = 0.
3. While (m < n - 1)

3a. Find an edge e in E of least weight.

3b. Delete e from E.

3c. If addition of e to the edge set of T does not produce a circuit then add e to the edge set of T and set m: = m + 1

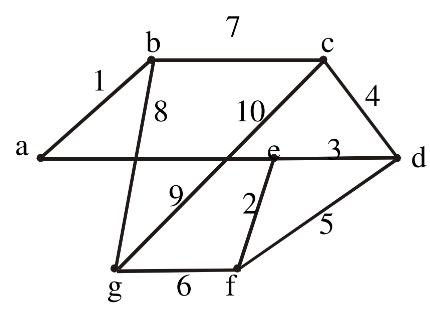
end while

Output T

end Algorithm

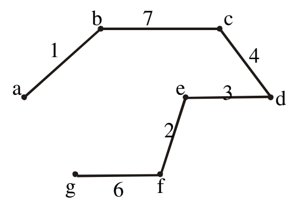
EXERCISE:

Use Kruskal's algorithm to find a minimal spanning tree for the graph below. Indicate the order in which edges are added to form the tree.



SOLUTION:

Minimal spanning tree:



Order of adding the edges:
{a,b}, {e,f}, {e,d}, {c,d}, {g,f}, {b,c}

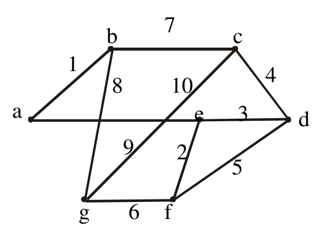
PRIM'S ALGORITHM:

Input: G[a weighted graph with n vertices]
Algorithm Body:
1. Pick a vertex v of G and let T be the graph with one vertex, v, and no edges.
2. Let V be the set of all vertices of G except v
3. for i: = 1 to n - 1
3a. Find an edge e of G such that
(1) e connects T to one of the vertices in V and
(2)e has the least weight of all edges connecting T to a vertex in V.
Let w be the end point of e that is in V.
3b. Add e and w to the edge and vertex sets of T and delete w from V.

next i
Output: T
end Algorithm

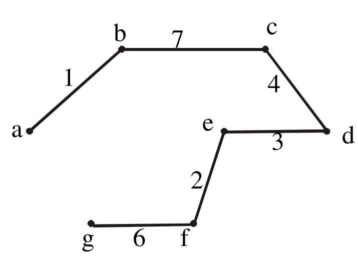
EXERCISE:

Use Prim's algorithm starting with vertex ***a*** to find a minimal spanning tree of the graph below. Indicate the order in which edges are added to form the tree.



SOLUTION:

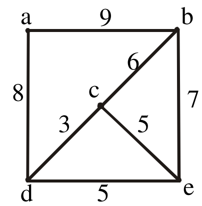
Minimal spanning tree is



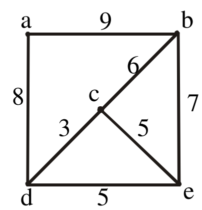
Order of adding the edges:
{a,b}, {b.c}, {c,d}, {d,e}, {e,f}, {f,g}

EXERCISE:

Find all minimal spanning trees that can be obtained using
(a) Kruskal's algorithm
(b) Prim's algorithm starting with vertex a

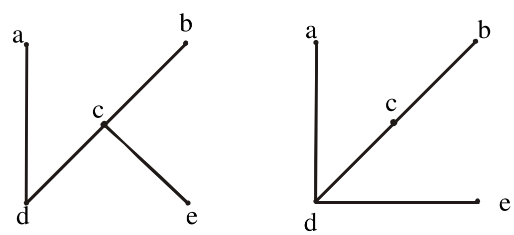


SOLUTION:
Given :

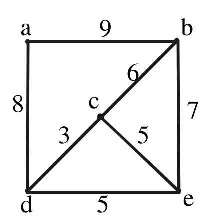


(a) When Kruskal's algorithm is applied, edges are added in one of the following two orders:
1. {c,d}, {c,e}, {c,b}, {d,a}
2. {c,d}, {d,e}, {c,b}, {d,a}

Thus, there are two distinct minimal spanning trees:



(b)



When Prim's algorithm is applied starting at a, edges are added in one of the following two orders:
1. {a,d}, {d,c}, {c,e}, {c,b}

2. {a,d}, {d,c}, {d,e}, {c,b}

Thus, the two distinct minimal spanning trees are: