Tree: Intro and Types

Friday, 26 December 2025 12:31 pm

https://mathigon.org/course/graph-theory/applications

In Computing & Technology

- **File Systems:** Hierarchical structure of folders and files (directories) on your computer.
- **Databases:** Indexing data for efficient searching (e.g., B-trees).
- **Compilers:** Parsing program code into syntax trees (Abstract Syntax Trees).
- **Search Engines & AI:** Indexing web pages, decision-making in games (game trees), and machine learning models (decision trees).
- **Networking:** Trie data structures for fast IP address lookups and routing.
- **Cryptography:** Merkle trees for verifying data integrity.

In Science & Engineering

- **Biology:** Modeling phylogenetic trees (evolutionary relationships) and analyzing DNA/RNA structures.
- **Hydrology:** Analyzing river networks and watersheds using Strahler stream order (a tree-based method).

- **Astronomy:** Space partitioning trees for organizing astronomical data (e.g., by NASA).

In Business & Everyday Life

- **Decision Making:** Decision trees for risk assessment, strategic planning, and evaluating options.
- **Organization:** Representing organizational charts, from company structures to family trees.
- **Logistics:** Optimizing delivery routes and supply chain flows (using graph/tree concepts).
- **Art & Design:** Representing fractal patterns in nature for aesthetic appreciation (e.g., "fractal bathing").

TREES

APPLICATION AREAS:

Trees are used to solve problems in a wide variety of disciplines. In computer science trees are employed to

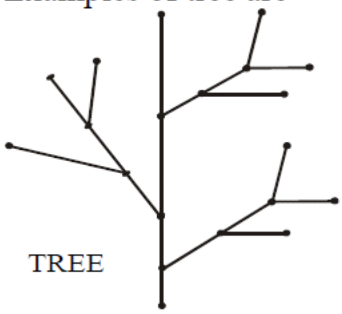
1) construct efficient algorithms for locating items in a list.
2) construct networks with the least expensive set of telephone lines linking distributed computers.
3) construct efficient codes for storing and transmitting data.
4) model procedures that are carried out using a sequence of decisions, which are valuable in the study of sorting algorithms.

TREE:

A tree is a connected graph that does not contain any non-trivial circuit. (i.e. it is circuit-free).

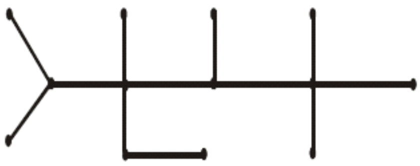
A trivial circuit is one that consists of a single vertex.

Examples of tree are



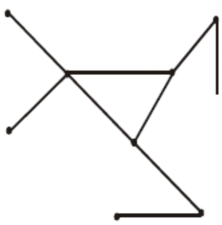
TREE

TREE



TREE

EXAMPLES OF NON TREES



(a) Graph with a circuit

(b) Disconnected graph

(c) Graph with a circuit

In graph (a), there exists circuit, so not a tree.
In graph (b), there exists no connectedness, so not a tree.
In graph (c), there exists a circuit (also due to loop), so not a tree (because trees have to be a circuit free).

SOME SPECIAL TREES

1. TRIVIAL TREE:

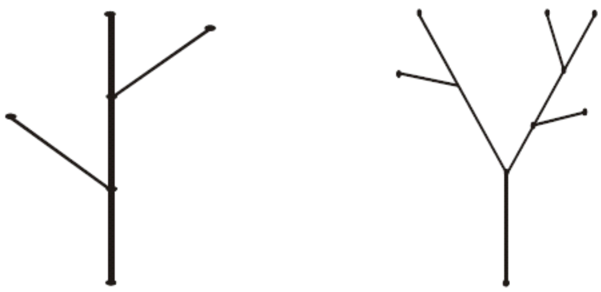
A graph that consists of a single vertex is called a trivial tree or degenerate tree.

2. EMPTY TREE

A tree that does not have any vertices or edges is called an empty tree.

3. FOREST

A graph is called a forest if, and only if, it is circuit-free.
OR "Any non-connected graph that contains no circuit is called a forest."
Hence, it clears that the connected components of a forest are trees.



A forest

PROPERTIES OF TREES:

1. A tree with n vertices has n - 1 edges (where $n \geq 0$).
2. Any connected graph with n vertices and n - 1 edges is a tree.
3. A tree has no non-trivial circuit; but if one new edge (but no new vertex) is added to it, then the resulting graph has exactly one non-trivial circuit.
4. A tree is connected, but if any edge is deleted from it, then the resulting graph is not connected.

5. Any tree that has more than one vertex has at least two vertices of degree 1.
6. A graph is a tree iff there is a unique path between any two of its vertices.

EXERCISE:

Explain why graphs with the given specification do not exist.
1. Tree, twelve vertices, fifteen edges.
2. Tree, five vertices, total degree 10.

SOLUTION:

1. Any tree with 12 vertices will have 12 - 1 = 11 edges, not 15.
2. Any tree with 5 vertices will have 5 - 1 = 4 edges.

Since, total degree of graph = 2 (No. of edges)
= 2(4) = 8

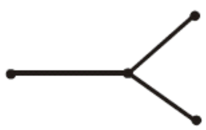
Hence, a tree with 5 vertices would have a total degree 8, not 10.

EXERCISE:

Find all non-isomorphic trees with four vertices.

SOLUTION:

Any tree with four vertices has (4-1=3) three edges. Thus, the total degree of a tree with 4 vertices must be 6 [by using total degree=2(total number of edges)].
Also, every tree with more than one vertex has at least two vertices of degree 1, so the only possible combinations of degrees for the vertices of the trees are 1, 1, 1, 3 and 1, 1, 2, 2.
The corresponding trees (clearly non-isomorphic, by definition) are



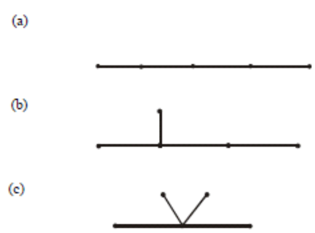
and

EXERCISE:

Find all non-isomorphic trees with five vertices.

SOLUTION:

There are three non-isomorphic trees with five vertices as shown (where every tree with five vertices has 5-1=4 edges).



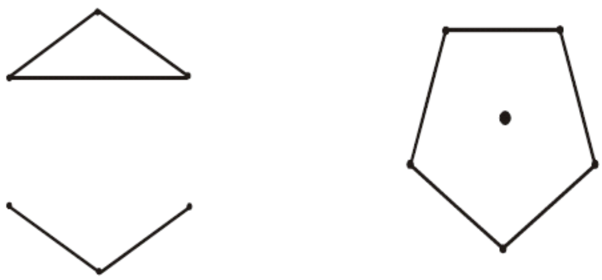
In part (a), tree has 2 vertices of degree '1' and 3 vertices of degree '2'.
In part (b), 3 vertices have degree '1', 1 has degree '2'and 1 vertex has degree '3'.
In part (c), possible combinations of degree are 1, 1, 1, 1, 4.

EXERCISE:

Draw a graph with six vertices, five edges that is not a tree.

SOLUTION:

Two such graphs are:



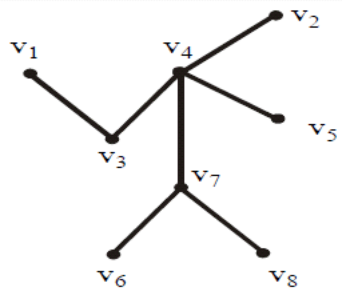
First graph is not a tree; because it is not connected also there exists a circuit. Similarly, second graph not a tree.

DEFINITION:

A vertex of degree 1 in a tree is called a ***terminal vertex*** or a leaf and a vertex of degree greater than 1 in a tree is called an ***internal vertex*** or a branch vertex.

EXAMPLE:

The terminal vertices of the tree are v_1, v_2, v_5, v_6 and v_8 and internal vertices are v_3, v_4, v_7.



ROOTED TREE:

A **rooted tree** is a tree in which one vertex is distinguished from the others and is called the ***root.***

The **level** of a vertex is the number of edges along the unique path between it and the root.

The **height** of a rooted tree is the maximum level to any vertex of the tree.

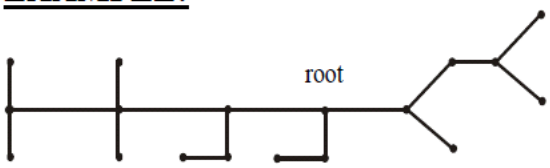
The **children** of any internal vertex v are all those vertices that are adjacent to v and are one level farther away from the root than v.

If **w** is a **child** of **v**, then **v** is called the **parent** of **w**.

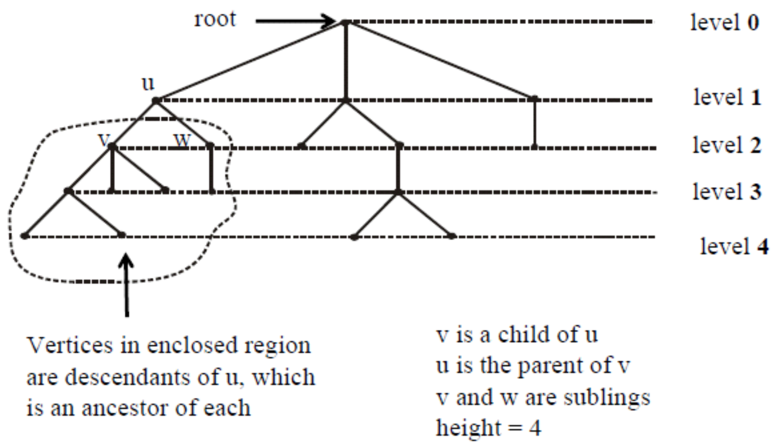
Two vertices that are both children of the same parent are called **siblings**.

Given vertices v and w, if v lies on the unique path between w and the root, then v is an **ancestor** of w and w is a **descendant** of v.

EXAMPLE:



We redraw the tree as and see what the relations are



EXERCISE:

Consider the rooted tree shown below with root v_0

a. What is the level of v_8?

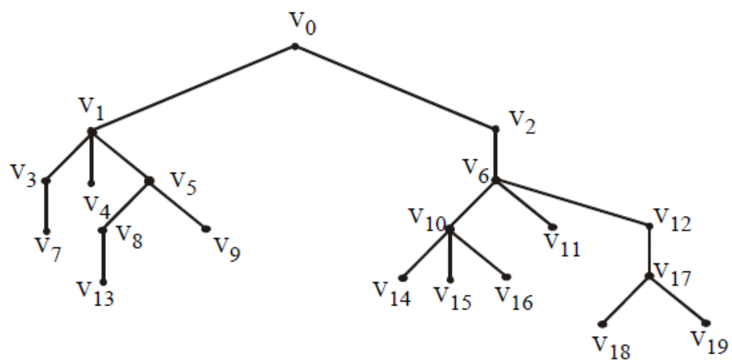
b. What is the level of v_0?

c. What is the height of this tree?

d. What are the children of v_{10}?

e. What are the siblings of v_1?

f. What are the descendants of v_{12}'



SOLUTION:

As we know that "Level means

the total number of edges along the unique path between it and the root".

(a). As v_0 is the root so the level of v_8 (from the root v_0 along the unique path) is 3,because it covers the 3 edges.

(b).The level of v_0 is 0(as no edge cover from v_0 to v_0).

(c).The height of this tree is 5.

Note: As levels are 0, 1, 2, 3, 4, 5 but to find height we have to take the maximum level.

(d).The children of v_{10} are v_{14}, v_{15} and v_{16} .

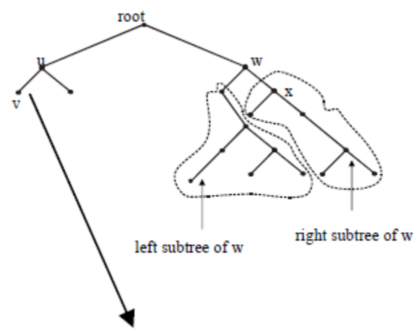
(e).The siblings of v_1 are v_3 , v_4 , and v_5.

(f).The descendants of v_{12} are v_{17}, v_{18}, and v_{19}.

BINARY TREE

A ***binary tree*** is a rooted tree in which every internal vertex has at most two children.
Every child in a binary tree is designated either a left child or a right child (but not both).
A ***full binary tree*** is a binary tree in which each internal vertex has exactly two children.

EXAMPLE:



v is the left child of u.

THEOREMS:

1. If k is a positive integer and T is a full binary tree with k internal vertices, then T has a total of 2k + 1 vertices and has k + 1 terminal vertices.

2. If T is a binary tree that has t terminal vertices and height h, then $t \le 2^h$
Equivalently,

$$\log_2 t \le h$$

Note: The maximum number of terminal vertices of a binary tree of height h is 2^h.

EXERCISE:

Explain why graphs with the given specification do not exist.
1. full binary tree, nine vertices, five internal vertices.
2. binary tree, height 4, eighteen terminal vertices.

SOLUTION:

1. Any full binary tree with five internal vertices has six terminal vertices, for a total of eleven vertices (according to 2(5) +1=11), not nine vertices in all.

OR

As total vertices = 2k+1=9
k=4(internal vertices)
but given internal vertices=5 ,which is a contradiction.
Thus there is no full binary tree with the given properties.

2. Any binary tree of height 4 has at most $2^4 = 16$ terminal vertices.
Hence, there is no binary tree that has height 4 and eighteen terminal vertices.

EXERCISE:

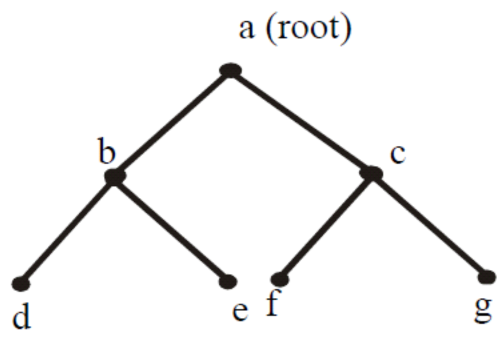
Draw a full binary tree with seven vertices.

SOLUTION:

Total vertices= 2k + 1 = 7 (by using the above theorem)

$\Rightarrow \quad k = 3$

Hence, total number of internal vertices (i.e. a vertex of degree greater than 1)=k=3
and total number of terminal vertices(i.e. a vertex of degree 1 in a tree) =k+1=3+1=4
Hence, a full binary tree with seven vertices is

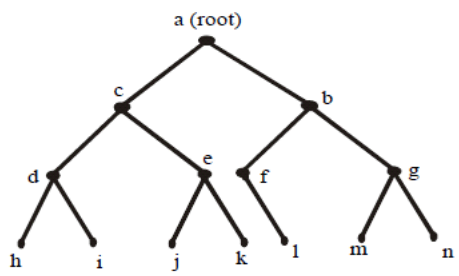


EXERCISE:

Draw a binary tree with height 3 and having seven terminal vertices.

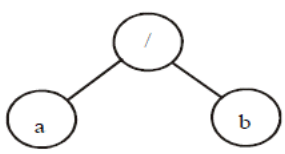
SOLUTION:

Given height=h=3
Any binary tree with height 3 has atmost 2^3=8 terminal vertices.
But here terminal vertices are 7
and Internal vertices=k=6 so binary tree exists and is as fellows:



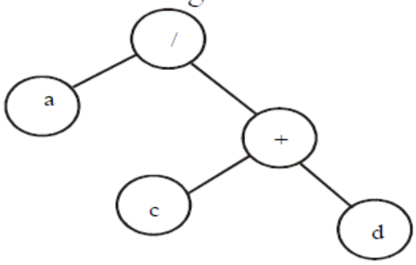
REPRESENTATION OF ALGEBRAIC EXPRESSIONS BY BINARY TREES

Binary trees are specially used in computer science to represent algebraic expression with Arbitrary nesting of balanced parentheses.



Binary tree for a/b

The above figure represents the expression a/b. Here the operator (/) is the root and b are the left and right children.



Binary tree for a/(c+d)

The second figure represents the expression a/(c+d).Here the operator (/) is the root. Here the terminal vertices are variables (here a, c and d), and the internal vertices are arithmetic operators (+ and /).

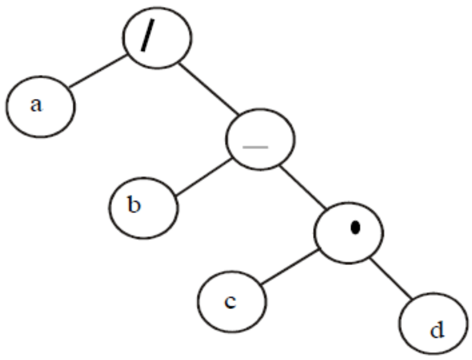
EXERCISE:

Draw a binary tree to represent the following expression

a/(b-c.d)

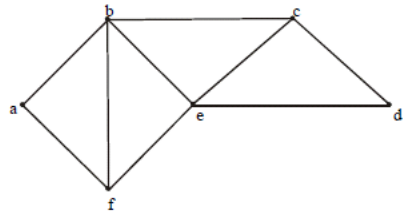
SOLUTION:

Note that the internal vertices are arithmetic operators, the terminal vertices are variables and the operator at each vertex acts on its left and right sub trees in left-right order.

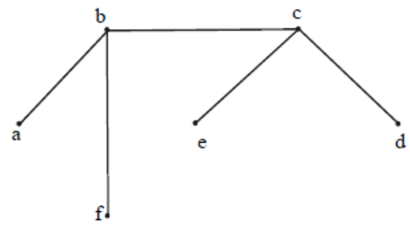


SPANNING TREES:

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:

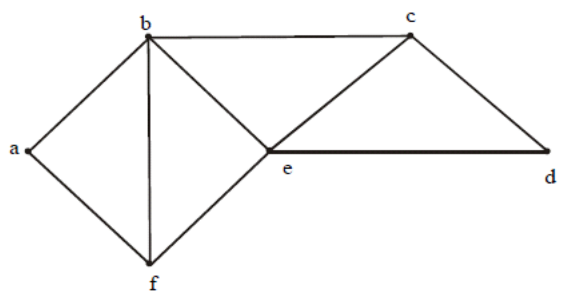
A spanning tree for a graph G is a subgraph of G that contains every vertex of G and is a tree.

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:



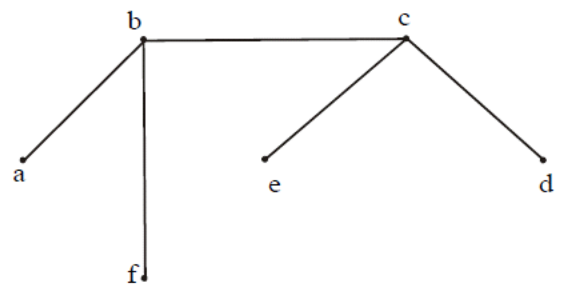
SOLUTION:

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

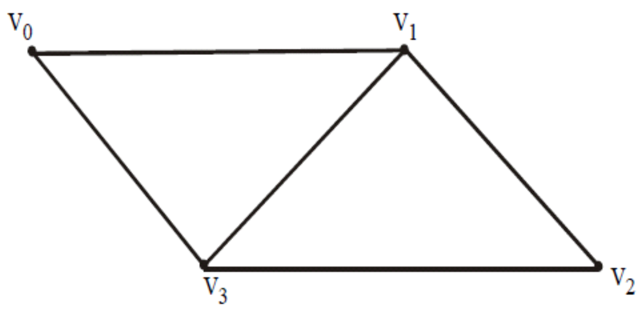
Note it that we can construct road from vertex a to b, but can't go from "a to e", also from "a to d" and from "a to c ", because there is no path available.

The associated spanning tree is



EXERCISE:

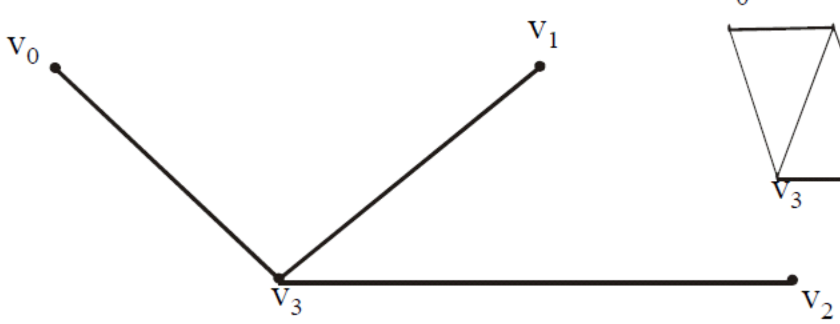
Find all the spanning trees of the graph given below.

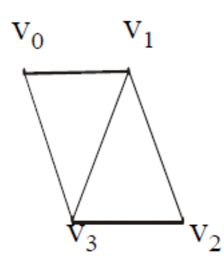


SOLUTION:

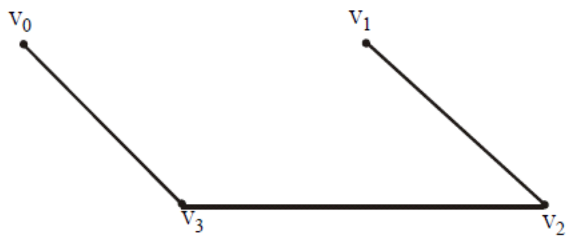
The graph has n = 4 vertices and e = 5 edges. So we must delete e - v + 1 = 5 - 4 + 1 = 2 edges from the cycles in the graph to obtain a spanning tree.

(1) Delete v_0v_1 & v_1v_2 to get

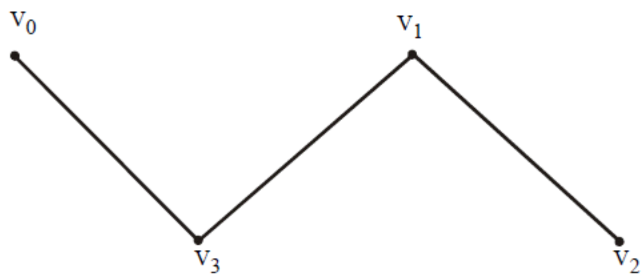


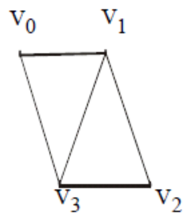


(2) Delete v_0v_1 & v_1v_3 to get

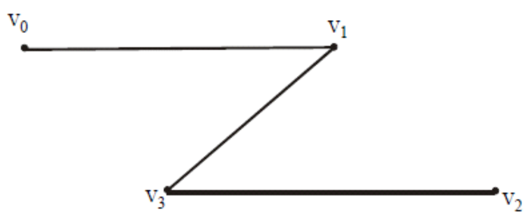


(3) Delete v_0v_1 & v_2v_3 to get

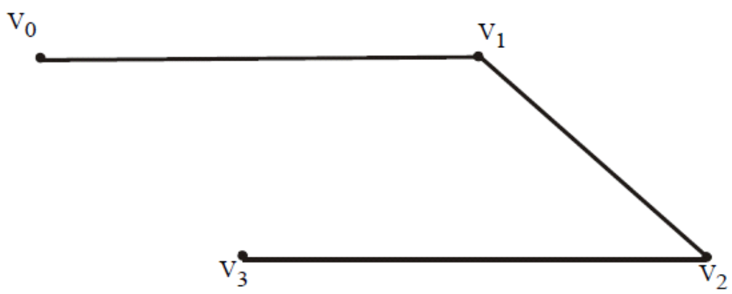


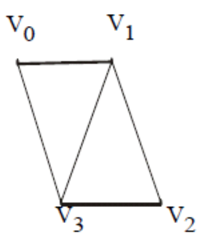


(4) Delete v_0v_3 & v_1v_2 to get

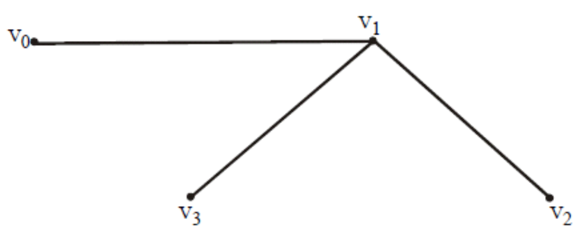


(5) Delete v_0v_3 & v_1v_3 to get

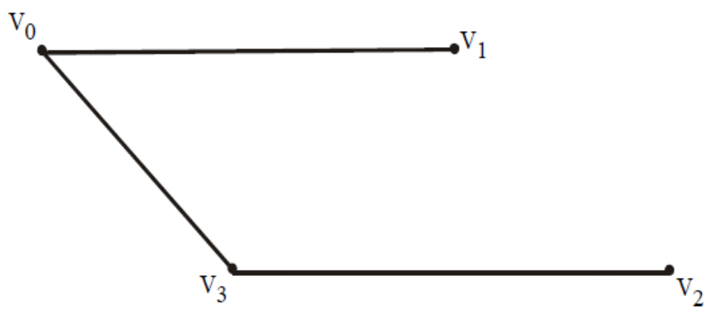


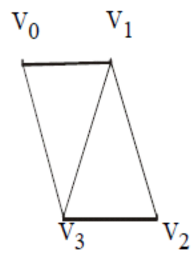


(6) Delete v_0v_3 & v_2v_3 to get

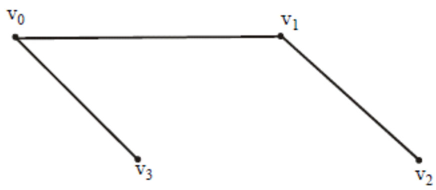


(7) Delete v_1v_3 & v_1v_2 to get





(8) Delete v_1v_3 & v_2v_3 to get



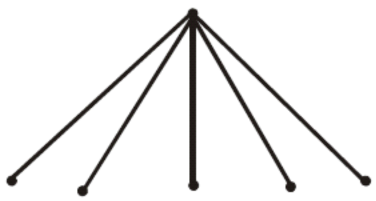
EXERCISE:

Find a spanning tree for each of the following graphs.

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

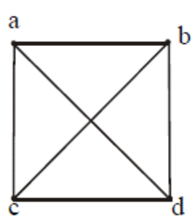
SOLUTION:

(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