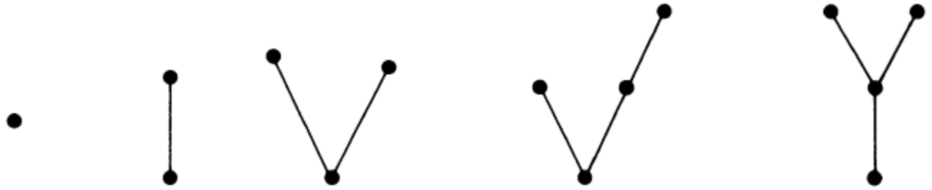
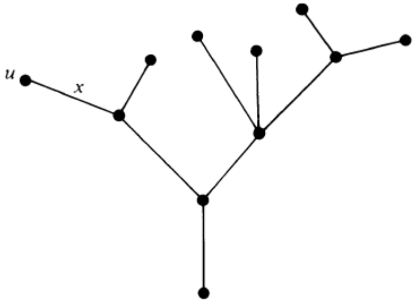
Tree:

A *tree* is a connected graph without any circuits.

Trees with one, two, three, and four vertices are shown in Fig.



Trees with one, two, three, and four vertices.



Tree.

- Every tree is a simple graph(i.e. without self loop and parallel edges.)
- The subgraph of tree is also a tree.

Example: Which of the graphs shown in Figure 2 are trees?

Solution: G_1 and G_2 are trees, because both are connected graphs with no simple circuits. G_3 is not a tree because e, b, a, d, e is a simple circuit in this graph. Finally, G_4 is not a tree because it is not connected. ◀

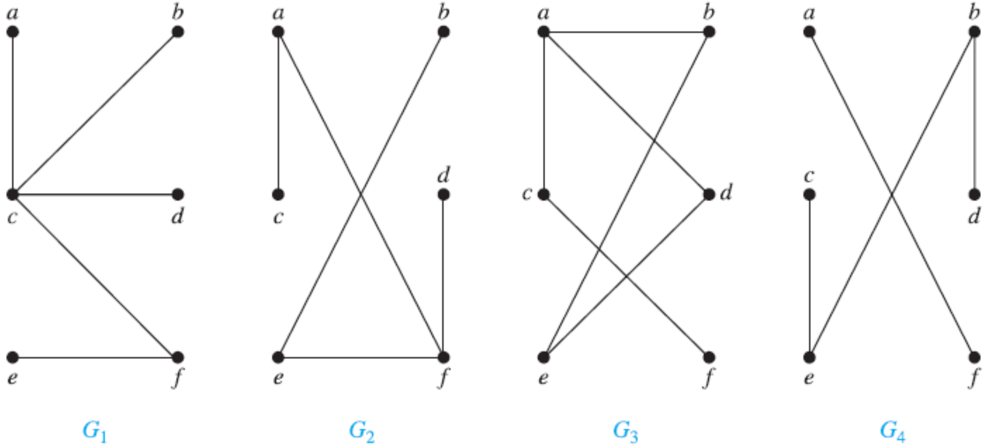


FIGURE 2 **Examples of Trees and Graphs That Are Not Trees.**

SOME PROPERTIES OF TREES

THEOREM

There is one and only one path between every pair of vertices in a tree, T.

Proof: Since T is a connected graph, there must exist at least one path between every pair of vertices in T. Now suppose that between two vertices a and b of T there are two distinct paths. The union of these two paths will contain a circuit and T cannot be a tree.

THEOREM

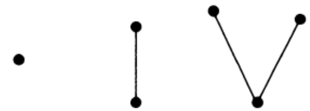
If in a graph G there is one and only one path between every pair of vertices, G is a tree.

Proof: Existence of a path between every pair of vertices assures that G is connected. A circuit in a graph (with two or more vertices) implies that there is at least one pair of vertices a, b such that there are two distinct paths between a and b. Since G has one and only one path between every pair of vertices, G can have no circuit. Therefore, G is a tree.

THEOREM

A tree with n vertices has $n-1$ edges.

Proof We prove the result by using induction on n, the number of vertices. The result is obviously true for $n = 1$, 2 and 3. (Fig. 1)



(Fig. 1)

Let the result be true for all trees with fewer than n vertices. Let T be a tree with n vertices and let e be an edge with end vertices u and v. So the only path between u and v is e. Therefore deletion of e from T disconnects T.

Now, $T-e$ consists of exactly two components T_1 and T_2 say, and as there were no cycles to begin with, each component is a tree.

Let n_1 and n_2 be the number of vertices in T_1 and T_2 respectively, so that $n_1+n_2=n$. Also, $n_1<n$ and $n_2<n$. Thus, by induction hypothesis, number of edges in T_1 and T_2 are respectively n_1-1 and n_2-1. Hence the number of edges in $T = n_1-1+n_2-1+1 = n_1+n_2-1 = n-1$. ❑

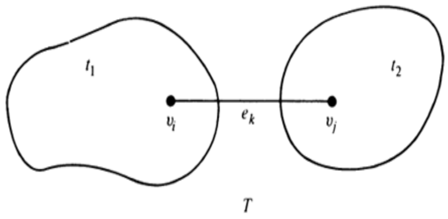


Fig.2 Tree T with n vertices.

Definition: A graph is said to be *minimally connected* if removal of any one edge from it disconnects the graph. Clearly, a minimally connected graph has no cycles.

Theorem A graph is a tree if and only if it is minimally connected.

Proof Let the graph G be minimally connected. Then G has no cycles and therefore is a tree.

Conversely, let G be a tree. Then G contains no cycles and deletion of any edge from G disconnects the graph. Hence G is minimally connected. ❑

Theorem A graph G with n vertices, $n-1$ edges and no cycles is connected.

Proof Let G be a graph without cycles with n vertices and $n-1$ edges. We have to prove that G is connected.

Assume that G is disconnected. So G consists of two or more components and each component is also without cycles. We assume without loss of generality that G has two components, say G_1 and G_2 (Fig. 4.1(b)).

Add an edge e between a vertex u in G_1 and a vertex v in G_2. Since there is no path between u and v in G, adding e did not create a cycle. Thus $G \cup e$ is a connected graph (tree) of n vertices, having n edges and no cycles. This contradicts the fact that a tree with n vertices has $n-1$ edges. Hence G is connected.

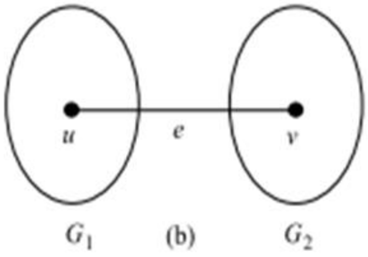


Fig. 4.1(b)

PENDANT VERTICES IN A TREE

Theorem In any tree (with two or more vertices), there are at least two pendant vertices.

Proof Let the number of vertices in a given tree T be $n (n > 1)$. So the number of edges in T is $n-1$. Therefore the degree sum of the tree is $2(n-1)$. This degree sum is to be divided among the n vertices. Since a tree is connected it cannot have a vertex of 0 degree. Each vertex contributes at least 1 to the above sum. Thus there must be at least two vertices of degree exactly 1.

Distance and Centre in a tree:

Distance:

In a tree, since there is exactly one path between any two vertices the determination of distance is much easier. For instance, in the tree of Fig. 3-7, $d(a, b) = 1$, $d(a, c) = 2$, $d(c, b) = 1$, and so on.

Eccentricity of a vertex:

The eccentricity $E(v)$ of a vertex v in a graph G is the distance from v to the vertex farthest from v in G; that is,

$$E(v) = \max_{v_i \in G} d(v, v_i).$$

In the given tree,

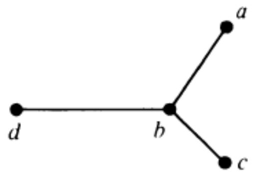


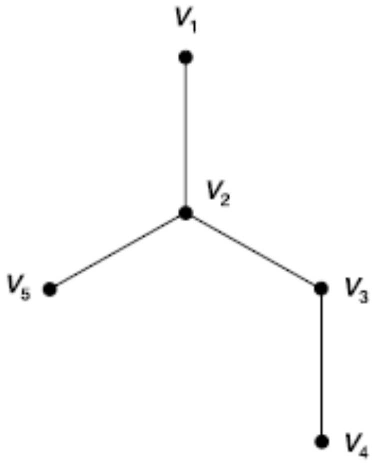
Fig. 3-7 Tree.

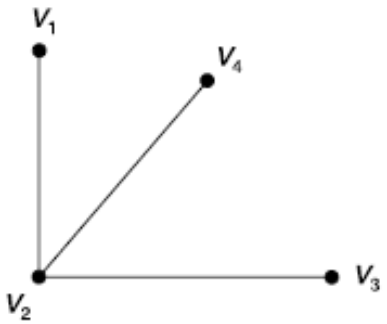
$d(a,b) = 1, d(a,c) = 2, d(a,d) = 2$
Maximum distance is 2. Therefore $E(a) = 2$.
Similarly $E(b) = 1, E(c) = 2, E(d) = 2$.

NOTE:

The maximum distance from each vertex of G occurs at a pendant vertices of G.

Ex: Find the eccentricity of v_1 in given Fig.





Centre of a tree:

A vertex with minimum eccentricity in graph G is called a *center* of G. The eccentricities of the four vertices in Fig. 3-7 are $E(a) = 2$, $E(b) = 1$, $E(c) = 2$, and $E(d) = 2$. Hence vertex b is the center of that tree.

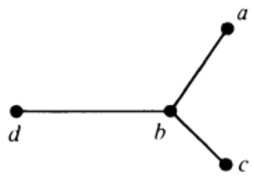
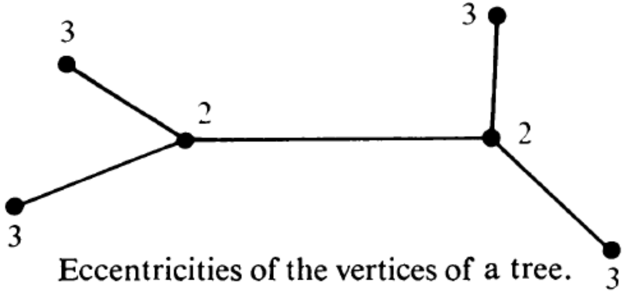


Fig. 3-7 Tree.

Consider the tree given below, the eccentricity of each vertex is shown next to the vertex. This tree has two vertices with minimum eccentricity. Therefore this tree has two centres.



Eccentricities of the vertices of a tree.

THEOREM

Every tree has either one or two centers.

Proof: The maximum distance, max $d(v, v_i)$, from a given vertex v to any other vertex v_i occurs only when v_i is a pendant vertex. With this observation, let us start with a tree T having more than two vertices. Tree T must have two or more pendant vertices .

Delete all the pendant vertices from T. The resulting graph T' is still a tree. The removal of all pendant vertices from T uniformly reduced the eccentricities of the remaining vertices (i.e., vertices in T') by one.

Therefore, all vertices that T had as centers will still remain centers in T'. From T' we can again remove all pendant vertices and get another tree T''. We continue this process (which is illustrated in Fig.) until there is left either a vertex (which is the center of T) or an edge (whose end vertices are the two centers of T). Thus the theorem. ■

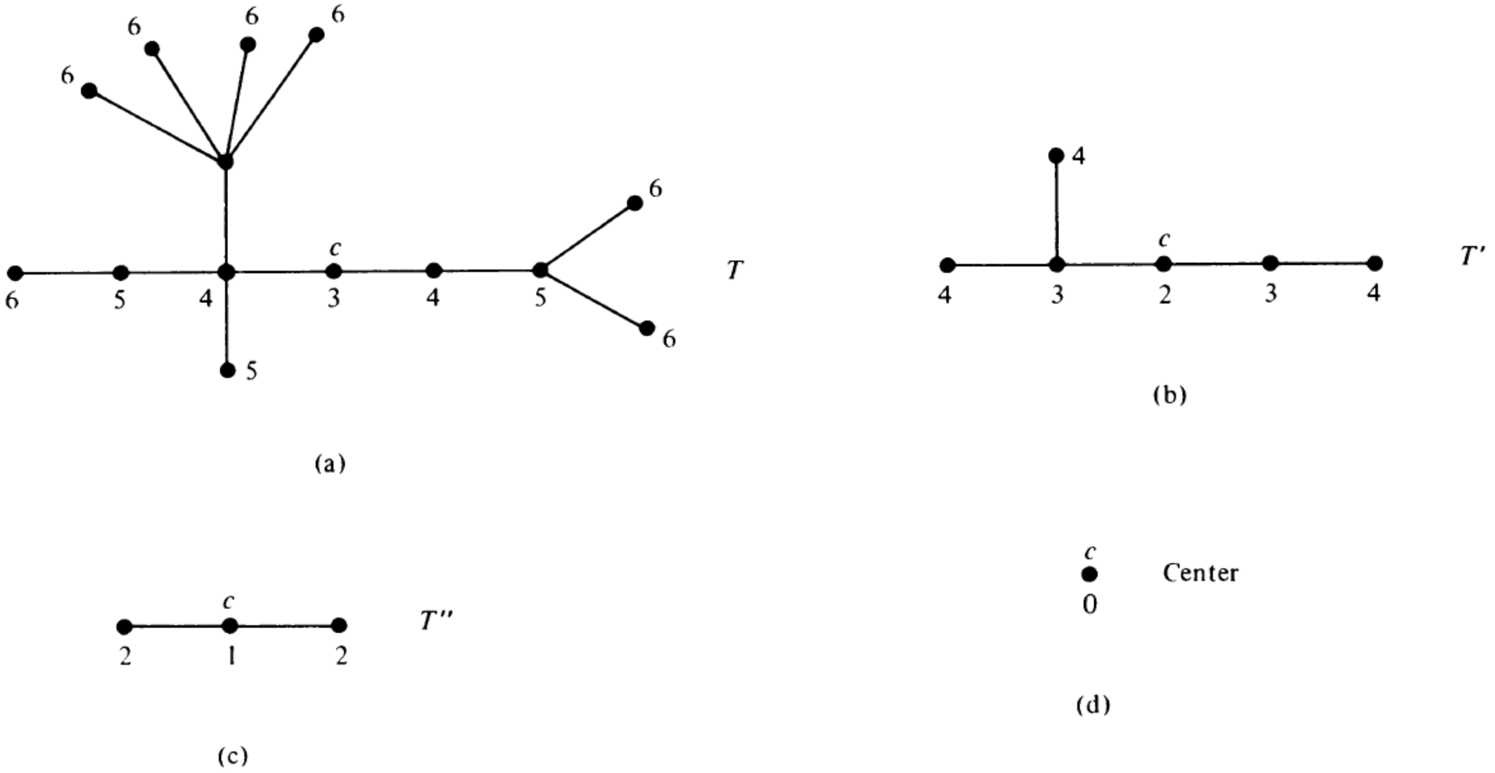


Fig. Finding a center of a tree.

Radius of a tree:

The eccentricity of the centre is called the radius of the tree.(i.e. minimum eccentricity among the vertices of G).
Radius(G)=$\min(E(v),\ v\ in\ V(G))$

Ex. Radius of given tree is: 1

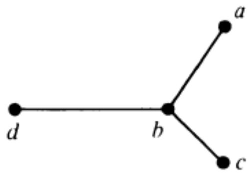


Fig. 3-7 Tree.

Diameter of a tree:

The diameter of G is the maximum eccentricity among the vertices of G.
Diameter (G)=$\max(E(v),\ v\ in\ V(G))$

Ex: diameter of given tree is: 2

Note: The radius of connected graph may not be half of its diameter.

Theorem: If r is the radius and d is the diameter of connected graph G then $r \leq d \leq 2r$.

Proof: From the definition of 'r' and 'd', we have $r \leq d$ ... (1)

Let u, v be the ends of a diametral path and w be the central vertex then

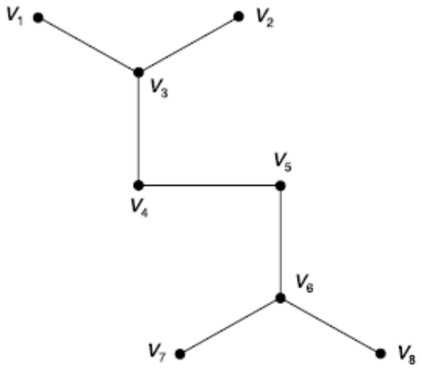
$$D = d(u, v) \leq d(u, w) + d(w, v) \leq 2r \qquad \text{(Triangle inequality)}$$

or
$$d \leq 2r \qquad \text{... (2)}$$

From (1) and (2)

$$r \leq d \leq 2r$$

Example : Consider the tree T shown in Fig.



We have

$$e(v_1) = 5,\ e(v_2) = 5,\ e(v_3) = 4,$$
$$e(v_4) = 3,\ e(v_5) = 3,\ e(v_6) = 4,$$
$$e(v_7) = e(v_8) = 5$$

The radius of $T = r = 3$
and the diameter of $T = 5$

Example : Consider the graph shown in Fig.

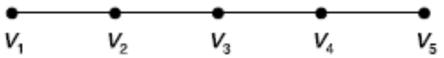


Fig.

$e(v_1) = 4$, $e(v_2) = 3$, $e(v_3) = 2$,

$e(v_4) = 3$, $e(v_5) = 4$, radius of $G = i = 2$.

Hence centre of $G = \{v_3\}$.

Example : In the graph shown in Fig.

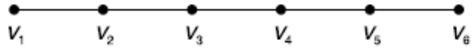


Fig.

Centre of $G = \{v_3, v_4\}$

Example : In the graph shown in Fig. $C(G) = \{v_4, v_5\}$:

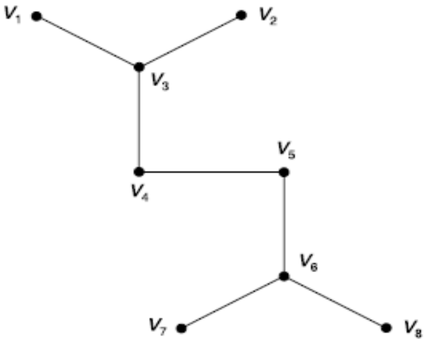


Fig.

Spanning Tree:

Let G be a connected graph, then the subgraph T of G is called a spanning tree of G if

(i) T is a tree and

(ii) it contains all the vertices of G.

Ex 1:

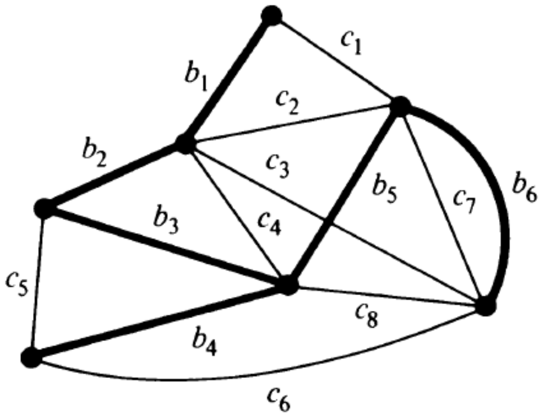
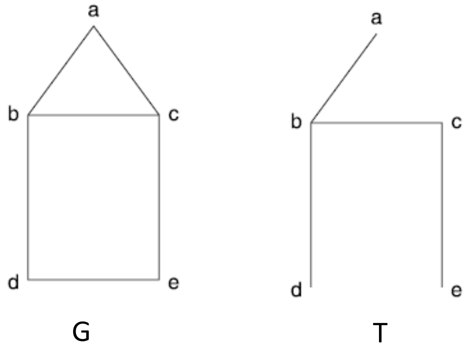


Fig. Spanning tree.

Ex 2:



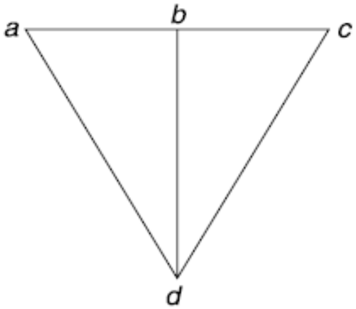
It is to be noted that a spanning tree is defined only for a connected graph, because a tree is always connected, and in a disconnected graph of n vertices we cannot find a connected subgraph with n vertices.

Each component (which by definition is connected) of a disconnected graph, however, does have a spanning tree. Thus a disconnected graph with k components has a *spanning forest* consisting of k spanning trees. (A collection of trees is called a *forest*.)

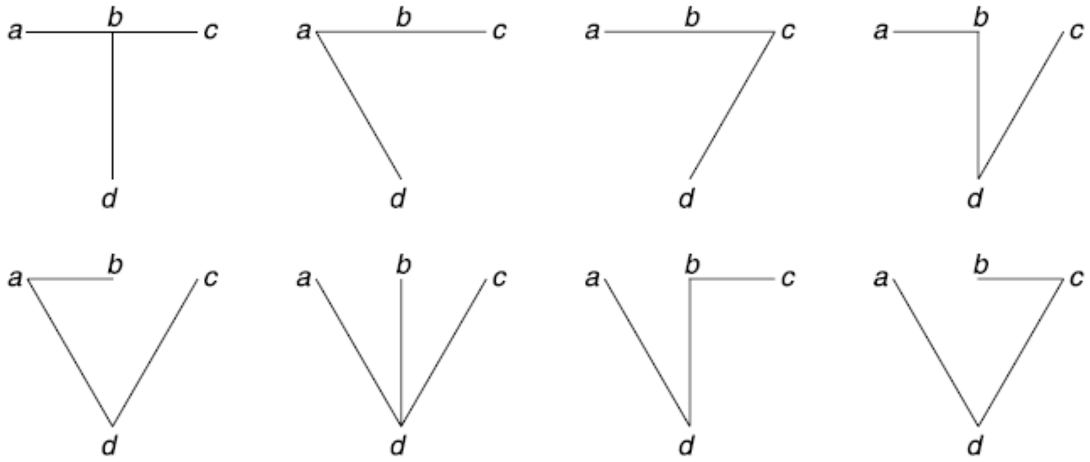
NOTE:

- Let G be a connected graph with n vertices and m edges. A spanning tree of G contains $(n-1)$ edges.
- Every connected graph has at least one spanning tree.
- A complete graph K_n has n^{n-2} different spanning trees.
- Since spanning trees are the largest(with maximum number of edges) trees among all tree in G, so it is called as a maximal tree subgraph or maximal tree of G.

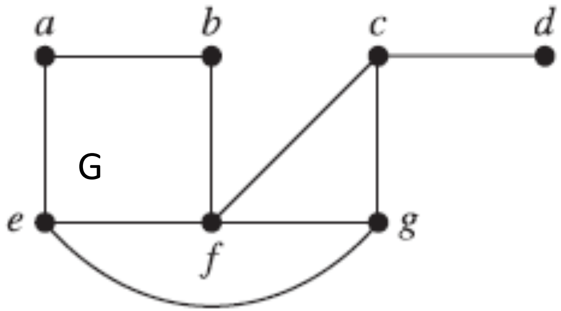
Example: Find all the spanning trees of the graph G shown in the Fig.



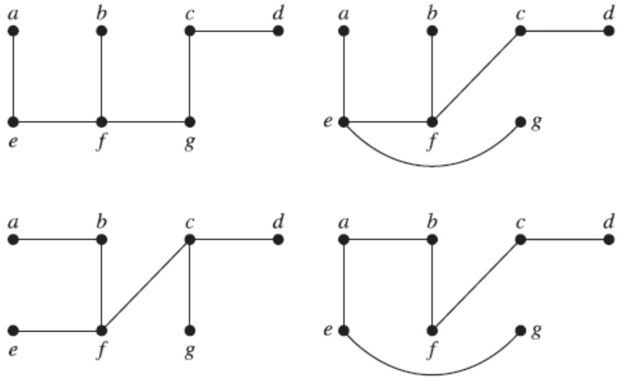
Solution: The spanning trees of G are given below



Ex: Find all the spanning trees in graph G.



Solution:



Spanning Trees of *G*.

Branch: An edge in a spanning tree T is called a branch of T.

Chord: An edge of G that is not in a given spanning tree is called a chord.

Branches and chords are defined with respect to given spanning tree. An edge which is a branch of one spanning tree T_1 may be a chord with respect to other spanning tree T_2.

edges b_1, b_2, b_3, b_4, b_5, and b_6 are branches of the spanning tree shown in Fig. while edges $c_1, c_2, c_3, c_4, c_5, c_6, c_7$, and c_8 are chords.

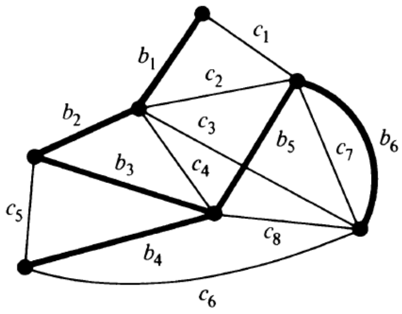


Fig. Spanning tree.

THEOREM

With respect to any of its spanning trees, a connected graph of n vertices and e edges has $n - 1$ tree branches and $e - n + 1$ chords.

For example, the graph in Fig. (with $n = 7$, $e = 14$), has six tree branches and eight chords with respect to the spanning tree $\{b_1, b_2, b_3, b_4, b_5, b_6\}$. Any other spanning tree will yield the same numbers.

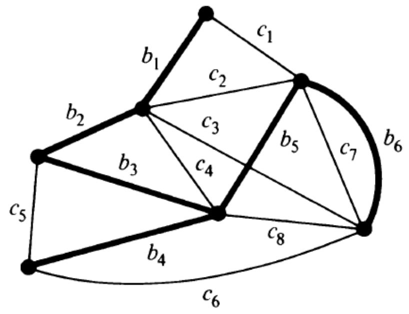


Fig. Spanning tree.

Rank and Nullity of a graph:

If a graph G has k component, n vertices and e edges then

Rank $r \ = n - k$
Nullity $\boldsymbol{\mu} = \ e - n + k$

If graph G is connected graph then $k = 1$,
Rank $r \ = n - 1$
Nullity $\boldsymbol{\mu} = \ e - n$ +1

It may be observed that

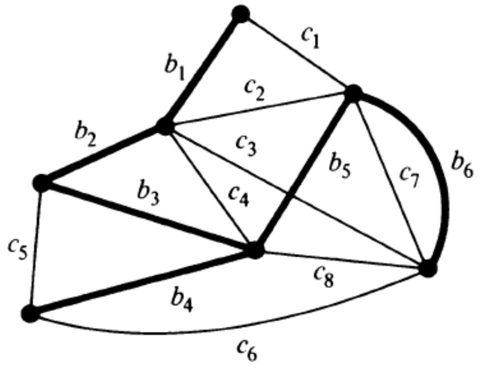
Rank of G= No of branches in spanning tree of G
Nullity of G= Number of chords in G
Rank + Nullity= Number of edges in G.

Fundamental Circuit:

Let T be a spanning tree in a connected graph G. Adding one chord to T will create exactly one circuit. Such a circuit , formed by adding a chord to a spanning tree is called Fundamental Circuit.

Note: Since spanning trees are the largest trees(with maximum number of edges), among all trees in G, it is called a maximal tree subgraph or maximal tree of G.

- A circuit is fundamental with respect to given spanning tree.



In the given spanning tree
The branches are $\{b_1, b_2, b_3, b_4, b_5, b_6\}$
The chords are $\{c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8\}$

Adding c_1 to spanning tree we get $\{b_1, b_2, b_3, b_4, b_5, b_6, c_1\}$ which has one circuit $\{b_1, b_2, b_3, b_5, c_1\}$, which is a fundamental circuit.
By adding c_2, we get fundamental circuit $\{b_2, b_3, b_5, c_2\}$,
By adding c_3, we get fundamental circuit $\{b_2, b_3, b_5, b_6, c_3\}$
Same way we can get other fundamental circuits.

- The number of fundamental circuits in a given spanning tree are $e - n + 1$ i.e. number of chords.

Fundamental Cut-Sets:

Consider a spanning tree T of a connected graph G. Take any branch b in T. Since $\{b\}$ is a cut-set in T, $\{b\}$ partitions all vertices of T into two disjoint sets-

(See graph: Two disjoints set are $\{v_1\}$ and $\{v_2, v_3, v_4, v_5, v_6\}$

Consider the same partition of vertices in G, and the cut set S in G that corresponds to this partition.

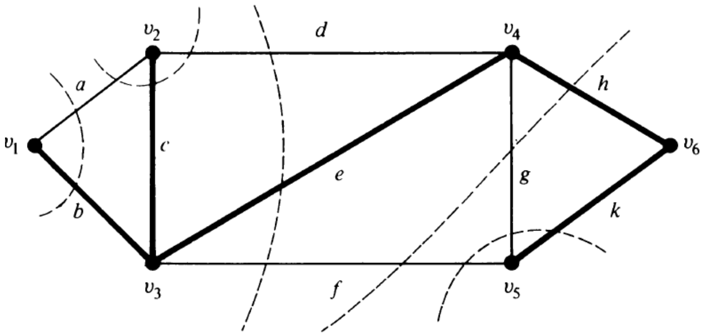
Cut- set S will contain only one branch b of T, and the rest (if any) of the edges in S are chords with respect to T.

Such a cut-set S containing exactly one branch of a tree T is called a *fundamental cut-set* with respect to T. fundamental cut-set is also called a *basic cut-set*.

In Fig., a spanning tree T (in heavy lines) and all five of the fundamental cut-sets with respect to T are shown (broken lines "cutting" through each cut-set).

Just as every chord of a spanning tree defines a *unique* fundamental circuit, every branch of a spanning tree defines a *unique* fundamental cut-set.

It must also be kept in mind that the term fundamental cut-set (like the term fundamental circuit) has meaning only with respect to a *given* spanning tree.



Fundamental cut-sets of a graph.

FINDING ALL SPANNING TREES OF A GRAPH

In a connected graph there are large number of spanning trees. One reasonable way to generate all the spanning trees of a graph is to start with a spanning tree (say a $b\ c\ d$ in given graph). Add a chord say h to the tree T_1. This forms a fundamental circuit $b\ c\ h\ d$.

Now removal of one edge say c from the fundamental circuit $b\ c\ h\ d$ will create new spanning tree.

This generation of one spanning tree from another is called cyclic interchange.

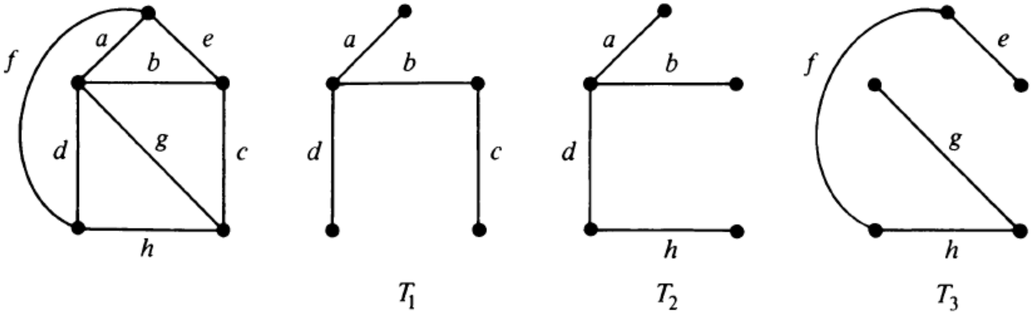
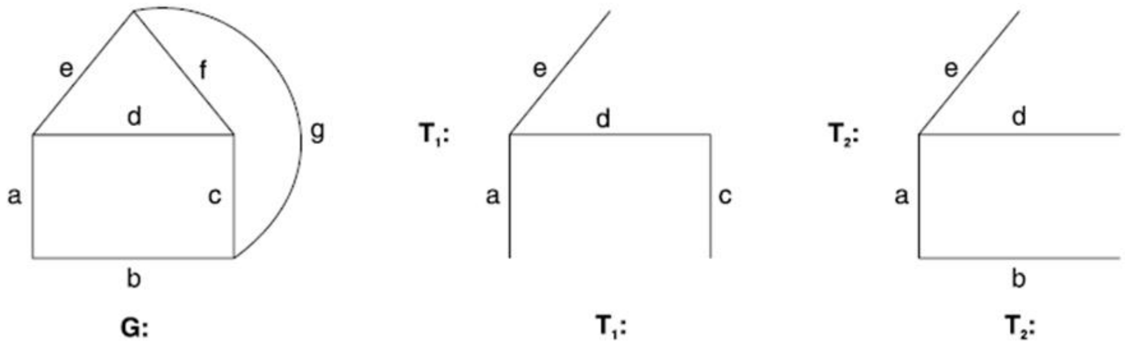


Fig. 3-19 Graph and three of its spanning trees.

DISTANCE BETWEEN SPANNING TREES OF A GRAPH

Let T_i and T_j be two spanning trees of a graph G. The distance between T_i and T_j is defined as the number of edges of G, present in T_i but not in T_j.

Example: Consider the graph G as shown below:



Graph G and two spanning trees T_1 and T_2.

The distance between the spanning trees T_1 and T_2 is one.

Spanning Tree in weighted graph:

Weight of a Spanning Tree:
The weight of a spanning tree T of a graph G is defined as the sum of the weights of all the branches in T.

Shortest Spanning Tree:
A spanning tree with smallest weight in a weighted graph is called a shortest spanning tree or shortest distance spanning tree or minimal spanning tree.

Algorithm for finding shortest spanning tree:

Prim's Algorithm: This algorithm based on vertex approach.

Let the graph G has n vertices v_1, v_2,…v_n. Tabulate the given weights of the edges of G in n by n table. Set the weights of non existing edges as very large i.e. ∞.

- Step 1:

  Start from any vertex, say v_1 and connect v_1 to its nearest neighbour(i.e. with the vertex which has smallest entry in row 1, say v_k)

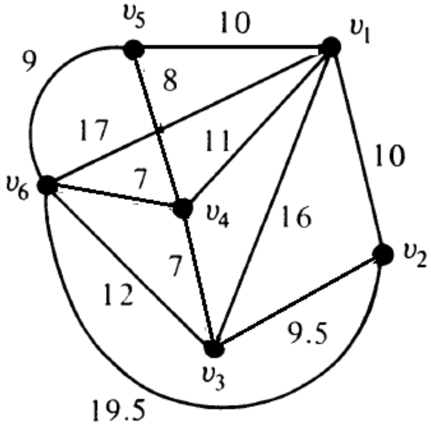
- Step 2:

  Consider $v_1 v_k$ as one subgraph, and connect this subgraph to its nearest neighbour(i.e. a vertex other than v_1 and v_k which has smallest entry among all the entries in rows 1 and k). Let this vertex be v_i.

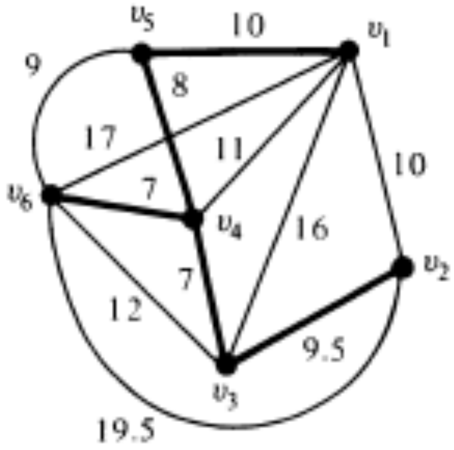
- Step 3:

  Next consider $v_1\ v_k\ v_i$ as one subgraph and continue the process until all the n vertices have been connected by $(n-1)$ edges.

Ex: Find shortest spanning tree in the graph given below:

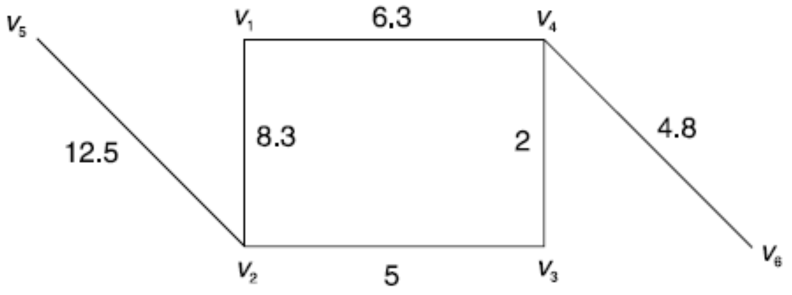


$$
\begin{array}{c|cccccc}
 & v_1 & v_2 & v_3 & v_4 & v_5 & v_6 \\
\hline
v_1 & - & 10 & 16 & 11 & 10 & 17 \\
v_2 & 10 & - & 9.5 & \infty & \infty & 19.5 \\
v_3 & 16 & 9.5 & - & 7 & \infty & 12 \\
v_4 & 11 & \infty & 7 & - & 8 & 7 \\
v_5 & 10 & \infty & \infty & 8 & - & 9 \\
v_6 & 17 & 19.5 & 12 & 7 & 9 & -
\end{array}
$$



Weight of shortest spanning tree is 10+8+7+7+9.5= 41.5

Ex: Find shortest spanning tree in the graph given below:



Let

$$e_1 = (v_1, v_2),\ e_2 = (v_2, v_3)$$
$$e_3 = (v_3, v_4),\ e_4 = (v_4, v_1)$$
$$e_5 = (v_2, v_5) \text{ and } e_6 = (v_4, v_6).$$

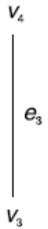
Denote the edge of G.

The weight of the edges are given by

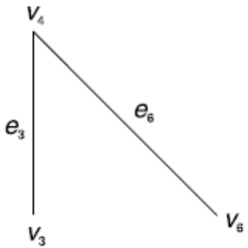
	v_1	v_2	v_3	v_4	v_5	v_6
v_1	0	8.3	∞	6.3	∞	∞
v_2	8.3	0	5	∞	12.5	∞
v_3	∞	5	0	2	∞	∞
v_4	6.3	∞	2	0	∞	4.8
v_5	∞	12.5	∞	∞	0	∞
v_6	∞	∞	∞	4.8	∞	0

We apply Prims algorithm to the graph as follows:

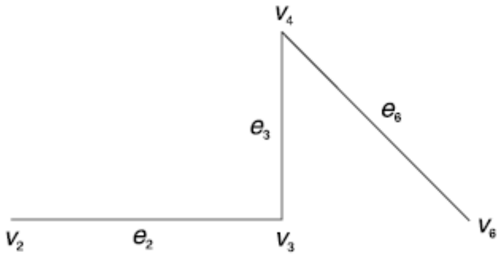
The edge $e_3 = (v_3, v_4)$ is an edge with minimum weight. Hence, we start with the vertex v_3 and select the edge e_3 incident with v_3.



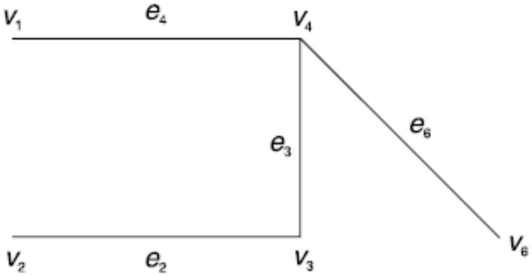
We next consider the edges connecting a vertex $\{v_3, v_4\}$ with the vertex of the set $V - \{v_3, v_4\}$. We observe that e_6 the edge with minimum weight.



Consider the edges connecting the vertices of the set $\{v_3, v_4, v_6\}$ with the vertices of $V - \{v_3, v_4, v_6\}$. The edge e_2 has the minimum weight. The edge e_2 is selected.



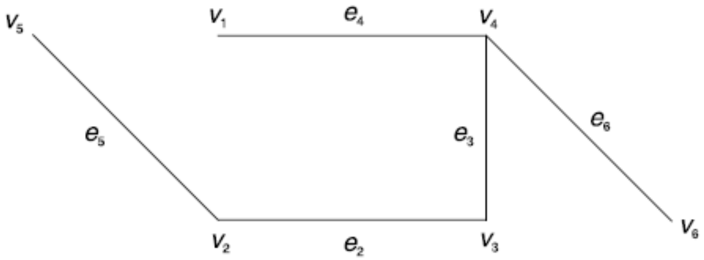
connecting the vertices of $\{v_2, v_3, v_4, v_6\}$; with the vertex set $V - \{v_2, v_3, v_4, v_6\}$, e_4 has minimum weight, therefore e_4 is selected.



e_1, e_5 are the edges remaining. e_5 is the only edge connecting $\{v_1, v_2, v_3, v_4, v_5, v_6\}$ and $\{v_5\}$ such that the inclusion of e_5 does not result in a cycle. Hence e_5 is selected.

Since number of edges selected is 5 we stop.

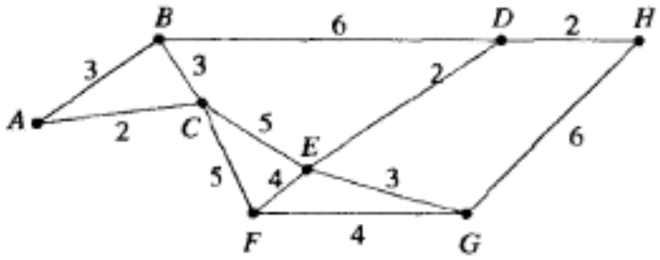
The minimal spanning tree obtained is shown in Fig. 8.106 (e).



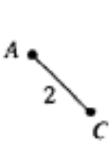
Weight of the minimal spanning tree

$$= 2 + 4.8 + 5 + 6.3 + 12.5$$
$$= 30.6$$

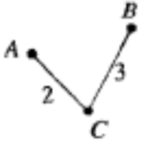
Ex: Find shortest spanning tree in the graph given below:



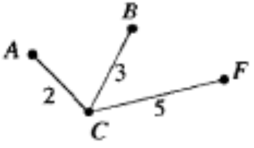
Solution:



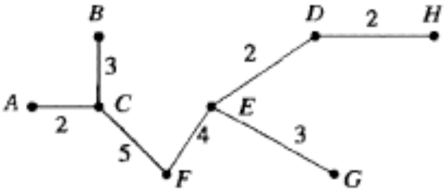
(a)



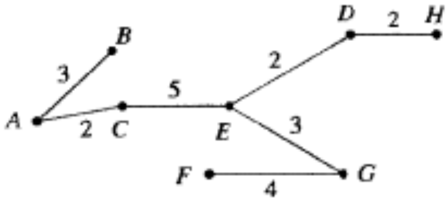
(b)



(c)

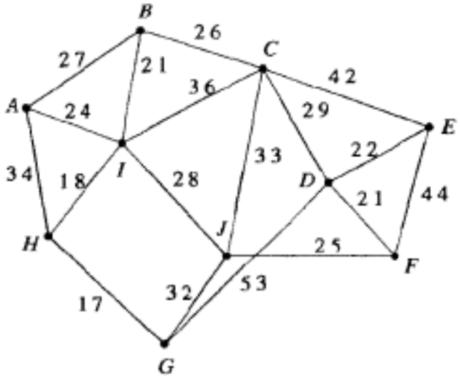


(d)

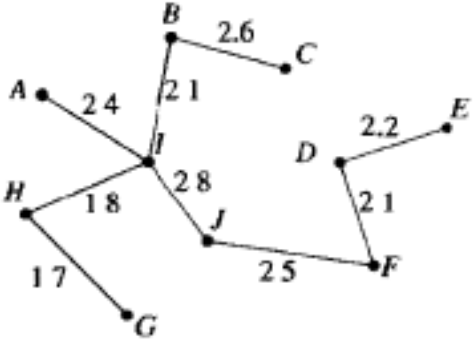


(e)

Ex: Find shortest spanning tree in the graph given below:



Solution:

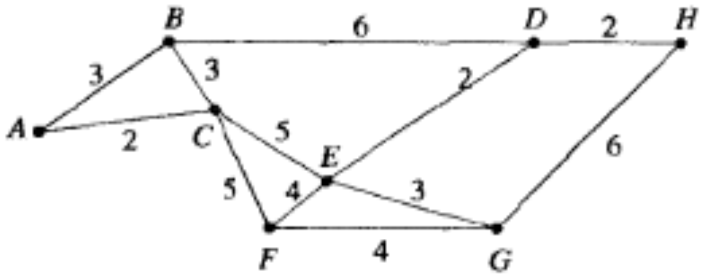


Kruskal Algorithm for finding shortest path:

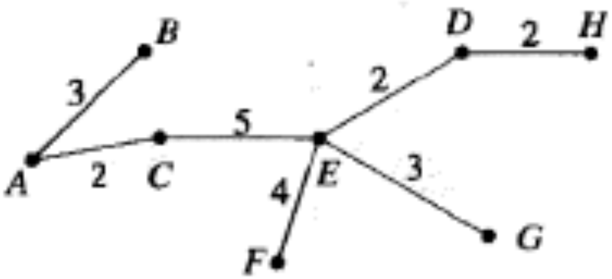
This algorithm is based on edge approach.

- List all the edges of the graph G in order of non increasing weights.
- Select, a smallest edge of G.
- For each successive step select another smallest edge(from all remaining edges of G) that makes no circuit with the previously selected edges.
- Continue until $(n-1)$ edges have been selected and these edges will constitute shortest spanning tree.

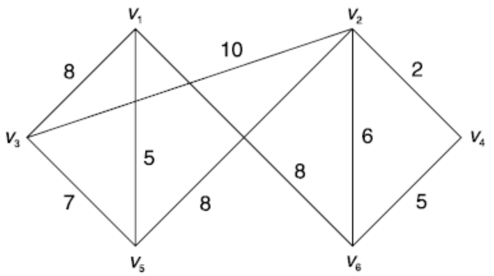
Ex: Find shortest spanning tree in the graph given below by kruskal’s algorithm:



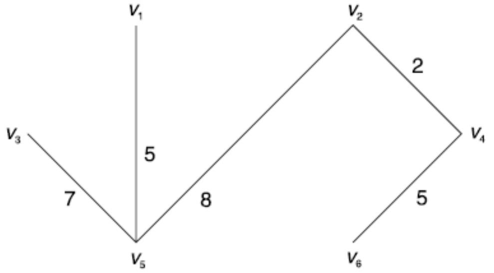
Solution:



Ex: Find shortest spanning tree in the graph given below by kruskal's algorithm:



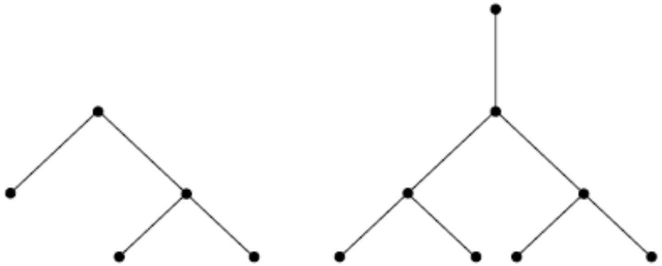
Solution: The minimal spanning tree of G is



The weight of the minimum spanning tree = 8 + 7 + 5 + 5 + 2 = 27.

Rooted tree: A tree in which one vertex (called the root) is distinguished from all the others is called a rooted tree.

Any tree may be made into a rooted tree by selecting one of the vertices as the root. A rooted is a directed tree if there is a root from which there is a directed path to each vertex of the tree. The graphs in Fig. are rooted trees in which the root of each is at the top.



Binary tree: A binary tree is defined as a tree in which there is exactly one vertex of degree two and each of the remaining vertices is of degree one or three.

Remark:

1. A node with degree two in a binary tree is said to be a root.
2. Every binary tree is a rooted tree.
3. A non-pendant vertex in a tree is called an internal vertex.
4. The number of internal vertices in a binary tree is one less than the number of pendant vertices

5. In a binary tree a vertex v_i is said to be at level l_i if v_i is at a distance of l_i from the root. Thus, the root is at level 0.

Theorem: The number of vertices in a binary tree is always odd.

Proof: Let T be a binary tree with n vertices.

By the definition of binary tree, one vertex is of degree two and remaining $(n-1)$ vertices are of degree one or three.

That is, the remaining $(n-1)$ vertices are of odd degree.

And since we know, the number of odd degree vertices in a graph is even in number, therefore

$(n-1)$ is even which implies n is odd

Tree traversal

A traversal a tree is a process to traverse (walk along) a tree in a systematic manner so that each vertex is visited and processed exactly once.

There are three methods of traversal of a binary tree

1. Preorder traversal
2. Inorder traversal, and
3. Postorder traversal

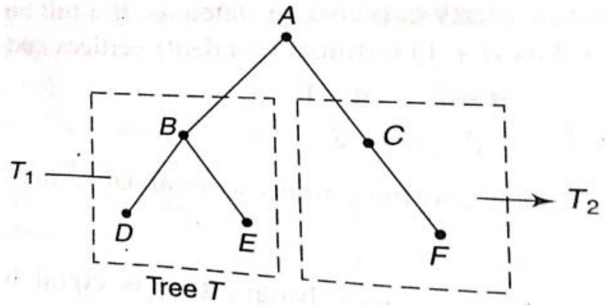
Preorder traversal: Let $T_1, T_2, \dots, T_n$ be the subtrees of the given binary tree T at the root R from left to right. The process of visiting the root R first and traversing T_1 in preorder then T_2 in preorder and so on until T_n is traversed in preorder is called the preorder traversal.

Example:

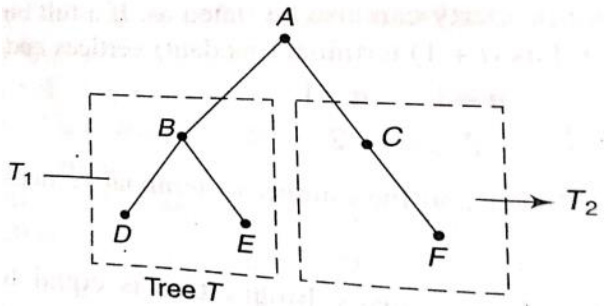
The preorder traversal of T visit the root A first and then traverse T_1 and T_2 in preorder.

The preorder traversal of T_1 visit the root B and then D and E in that order. The preorder traversal of T_2 visits the root C and then F

Thus, the preorder traversal of T is $\mathrm{AB}DECF$.



Inorder traversal: The process of traversing T_1 first in inorder and then visiting the root R and continuing the traversal of T_2 in inorder, T_3 in inorder etc. until T_n is traversed in inorder is called the inorder traversal.



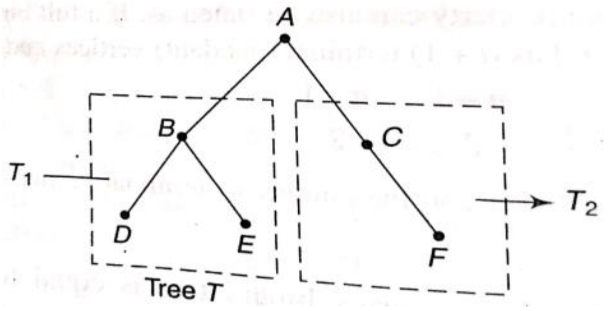
Example:

The inorder traversal of T traverses T_1 in inorder first, then visit the root A and finally traverse T_2 in inorder.

But the inorder traversal of T_1 processes D, B and E in that order and the inorder traversal of T_2 processes C and then F.

Thus, the inorder traversal of T is $DBEACF$.

Postorder traversal: The process of traversing T_1 first in post order and then T_2 in postorder, T_3 in postorder etc., T_n in post order and finally visiting the root R is called the postorder traversal.



Example:

The postorder traversal of T processes T_1, then T_2 in postorder and finally visits A.

But the postorder traversal of T_1 processes D, E and B in that order and the postorder traversal of T_2 processes F and then C.

Thus, the postorder traversal of T is $DEBFCA$.

Coloring, Covering and Partitioning

Proper Coloring:

Painting all the vertices of a graph with colors such that no adjacent vertices have same color, is called *proper coloring*.
A graph in which every vertex has been assigned a color according to proper coloring is called *properly colored graph*.
A graph can be properly colored in many different ways.

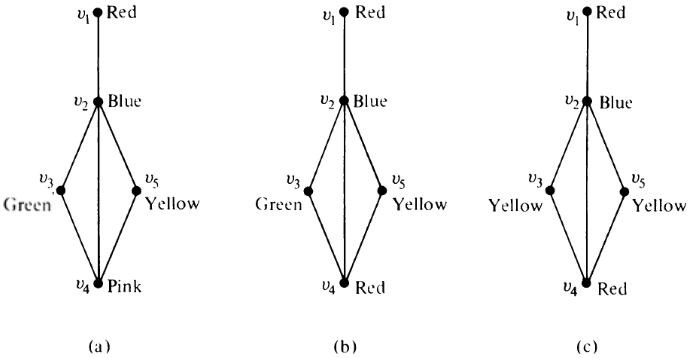


Fig. Proper colorings of a graph.

In Fig. $(a), (b), (c)$ number of colors that has been used for proper coloring are 5,4,3 respectively.
Hence the proper coloring is one that requires the minimum number of colors.

Chromatic Number:

The least number of different colors required for proper coloring of the graph is called Chromatic number of the graph. Chromatic number of a graph G is denoted by **χ**(G).

If the graph requires κ *different colors* for its proper coloring and no less, then κ is called chromatic number. i.e. **χ**(G)= κ.

Chromatic graph:

The graph having chromatic number κ is called κ-Chromatic graph.

Ex: Given graph is 3-Chromatic graph.

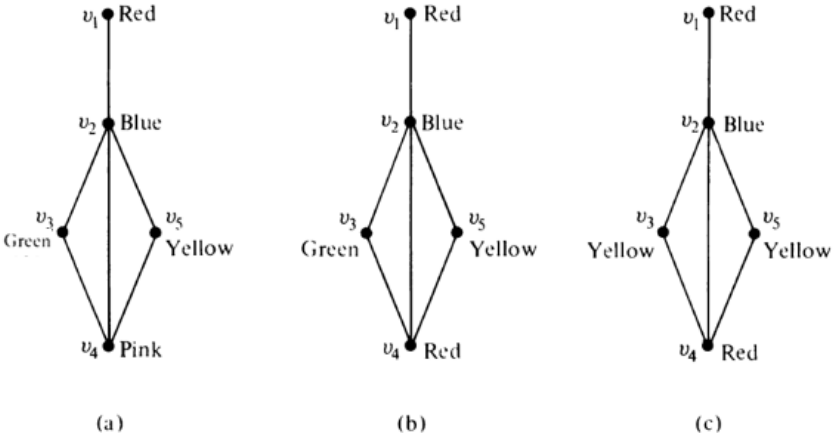
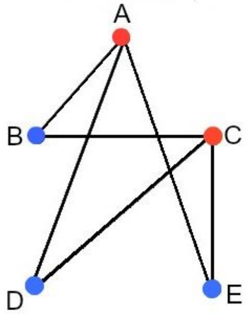
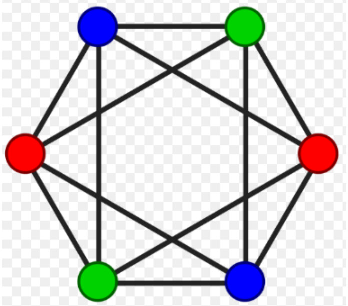


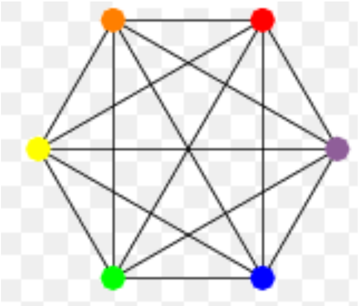
Fig. Proper colorings of a graph.



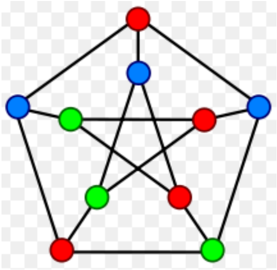
$\chi(G) = 2$
2-chromatic graph



$\chi(G) = 3$
3-chromatic graph



$\chi(G) = 6$
6-chromatic graph



$\chi(G) = 4$
4-chromatic graph

- **Some observation from Proper Coloring of graphs:**

1. A graph consist of only one vertex is 1-Chromatic.

2. A graph consist of one or more edges(not self loop) is at least 2-Chromatic.

3. A complete graph with n vertices is n-Chromatic as all its vertices are adjacent.

4. A graph containing a complete graph of r vertices is at least r-Chromatic. For ex: every graph containing a triangle is at least 3-Chromatic.

5. A graph consisting of simply one circuit with $n \geq 3$ vertices is **2-Chromatic, if n is even** and **3-Chromatic, if n is odd**.

Theorem: **Every tree with two or more vertices is 2-chromatic. Is converse true.**

Proof: Let T be a tree with root node v. Paint v with color 1. Paint all the vertices adjacent to v with color 2.

Next, paint the vertices adjacent to these (those that just have been colored with 2) using color 1. Continue this process till every vertex in T has been pained. Now in T we find that all vertices at odd distances from have color 2, while v and vertices at even distances from v have color 1.

Now along any path in T the vertices are of alternating colors. Since there is one and only path between every pair of vertices in a tree, no two adjacent vertices have same color.

Thus T has been properly colored with two colors.

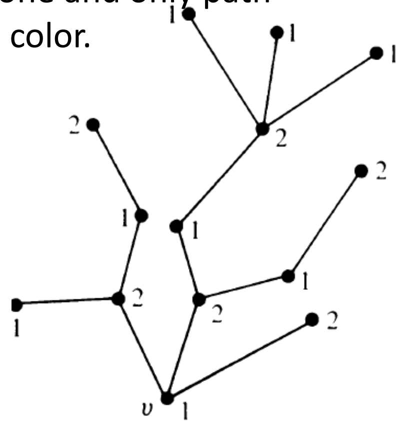
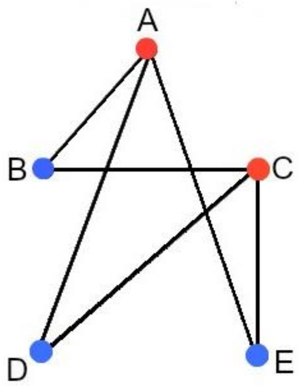


Fig. Proper coloring of a tree.

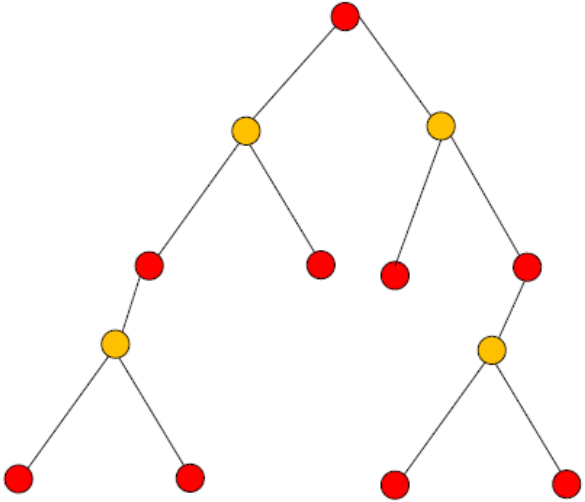
Every tree is 2-chromatic but not every 2-chromatic graph is tree.

Given graph is 2-chromaic but its not a tree.



$\chi(G) = 2$
2-chromatic graph

Chromatic Number of tree

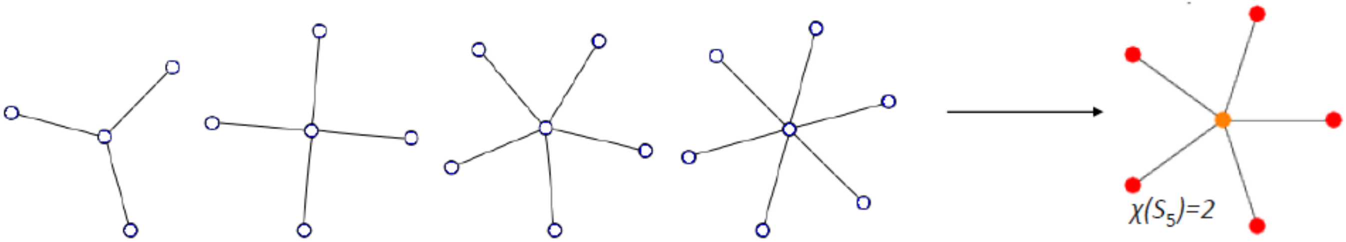


$\chi(G)=2$

So Chromatic Number of tree is 2

Chromatic number of Star Graph

- A star S_k is the complete bipartite graph $K_{1,k}$: a tree with one internal node and k leaves.



The star graphs S_3, S_4, S_5 and S_6.

For all star graph chromatic number is: $\chi(G)=2$

Theorem

A graph with at least one edge is 2-chromatic if and only if it has no circuits of odd length.

Theorem

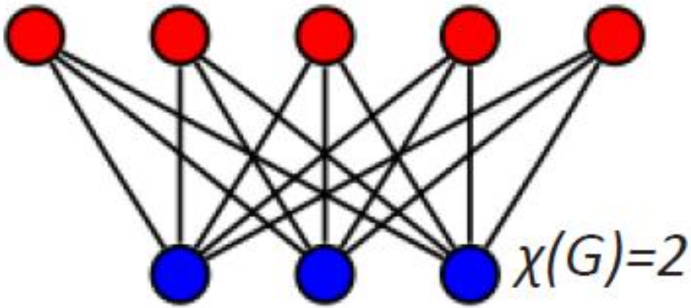
If d_{max} is the maximum degree of the vertices in a graph G,

$$\text{chromatic number of } G \leq 1 + d_{max}.$$

Chromatic Number of **Bipartite Graph**

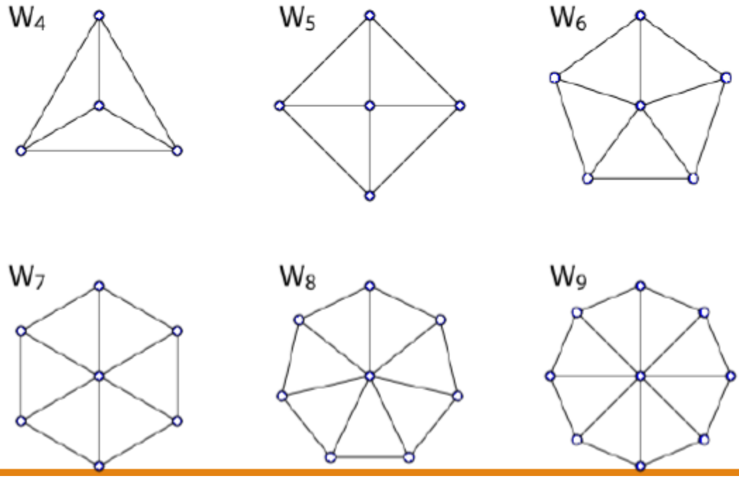
- A bipartite graph, also called a bigraph, is a set of graph vertices decomposed into two disjoint sets such that no two graph vertices within the same set are adjacent.

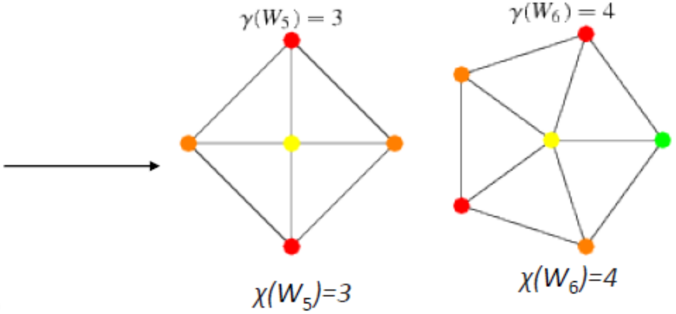
Clearly, every 2-chromatic graph is bipartite because the coloring partitions the vertex set into two subsets V_1 and V_2 such that no two vertices in V_1 (or V_2) are adjacent. Similarly, every bipartite graph is 2-chromatic.



Chromatic Number of Wheel Graph

- A wheel graph is a graph formed by connecting a single vertex to all vertices of a cycle.





A proper coloring of a graph naturally induces a partitioning of the vertices into different subsets. For example, the coloring in Fig. 1 produces the partitioning $\{v_1, v_4\}$, $\{v_2\}$, and $\{v_3, v_5\}$.

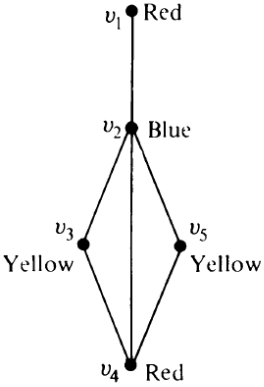


Fig.1

No two vertices in any of these three subsets are adjacent. Such a subset of vertices is called an independent set.

Independent set:

A set of vertices in a graph is said to be an *independent set* of vertices or simply an *independent set* if no two vertices in the set are adjacent.

$\{a, e\}, \{a, c, d\}, \{a, c, g\}, \{a, d, f\}, \{b, f\}, \{b, g\}$ etc **are independent sets.**

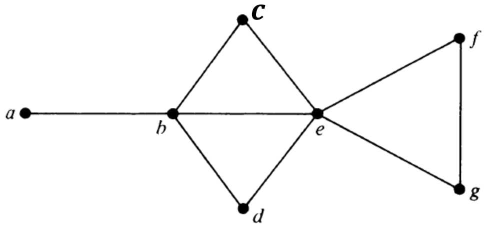


Fig. 2

Maximal Independent Set:

A *maximal independent set* (or *maximal internally stable set*) is an independent set to which no other vertex can be added without destroying its independence property. The set $\{a, c, d, f\}$ in Fig. is a maximal independent set. The set $\{b, f\}$ is another maximal independent set. The set $\{b, g\}$ is a third one.

From the example, it is clear that a graph, in general, has many maximal independent sets; and they may be of different sizes.

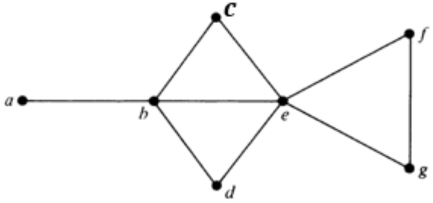


Fig. 2

Largest Independent Set:

A maximal independent set with largest number of vertices is called largest independent set.

In the given graph, the largest independent sets are $\{a, c, d, f\}, \{a, c, d, g\}$.

Independence Number:

The number of vertices in the largest independent set of a graph is called the independence number (or coefficient of internal stability), denoted by $\boldsymbol{\beta}(\mathrm{G})$.

The largest independent sets are $\{a, c, d, f\}, \{a, c, d, g\}$.
Hence $\boldsymbol{\beta}(\mathrm{G})=4$.

The largest independent sets are $\{a, c, d, f\}, \{a, c, d, g\}$.
Hence $\beta(G)=4$.

Note:

Consider a κ-chromatic graph G of n vertices properly colored with κ different colors. Since the largest number of vertices in G with the same color cannot exceed the independence number $\beta(G)$, we have the inequality

$$\beta(G) \geq \frac{n}{\kappa} \cdot$$

Note: 1. While using proper coloring of a graph the vertex set with same color of vertices is also an independent set.

2. To find chromatic number of graph G, we must find the minimum number of maximal independent sets, which collectively include all the vertices of G.
Ex: In the given graph $\{a, c, d, f\}$, $\{b, g\}, \{a, e\}$ are containing all the vertices of G. Thus the graph is three chromatic.

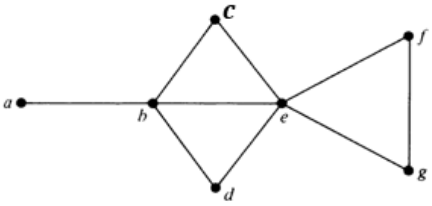


Fig. 2

Chromatic Partitioning: Given a simple, connected graph G, partition all vertices of G into the smallest possible number of disjoint, independent sets. This problem, known as the *chromatic partitioning* of graphs, is perhaps the most important problem in partitioning of graphs.

The following are some chromatic partitioning of graph G

$$\{(a, c, d, f), (b, g), (e)\},$$

$$\{(a, c, d, g), (b, f), (e)\},$$

$$\{(c, d, f), (b, g), (a, e)\},$$

$$\{(c, d, g), (b, f), (a, e)\}.$$

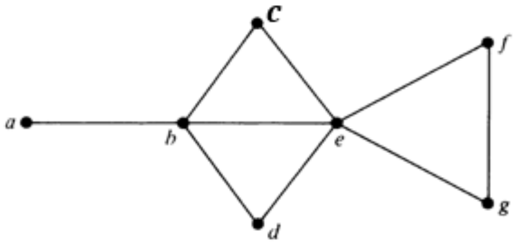
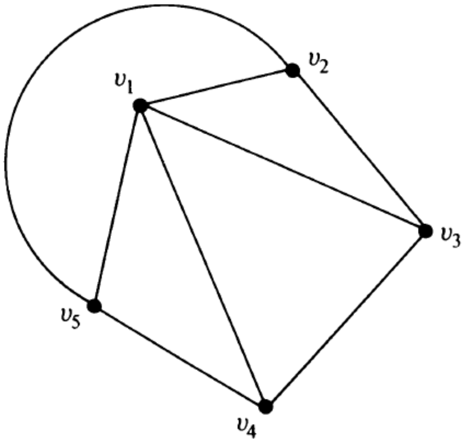


Fig. 2

The above graph doesn't have unique chromatic partitioning.

Uniquely Colorable Graphs: A graph that has only one chromatic partition is called a *uniquely colorable* graph.



A 3-chromatic graph.

The given graph has only one (unique) chromatic partitioning

$\{v_1\}$, $\{v_2, v_4\}$,$\{v_3, v_5\}$

Fusion: A pair of vertices a, b in a graph are said to be *fused* (*merged* or *identified*) if the two vertices are replaced by a single new vertex such that every edge that was incident on either a or b or on both is incident on the new vertex. Thus fusion of two vertices does not alter the number of edges, but it reduces the number of vertices by one. See Fig. 2-16 for an example.

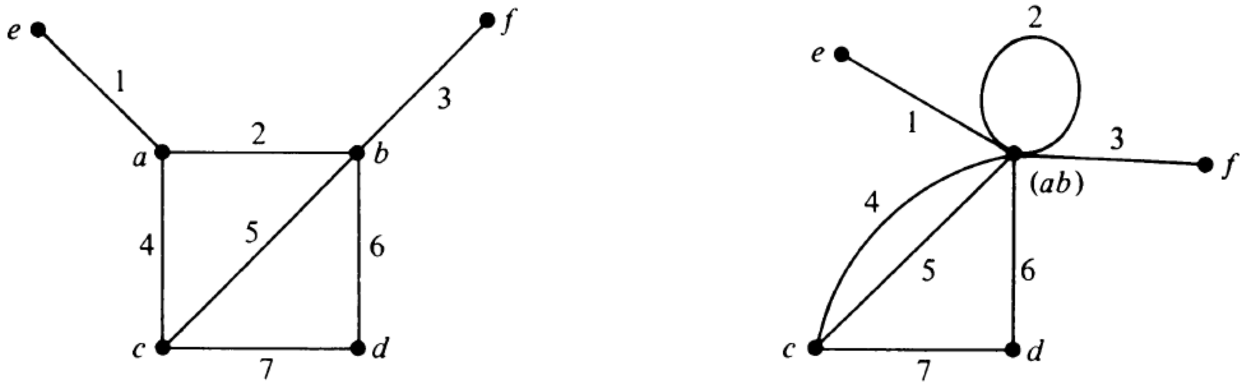


Fig. 2-16 Fusion of vertices a and b.

Chromatic polynomial

A given graph G of n vertices can be properly colored in many different ways using a sufficiently large number of colors. This property of graph can be expressed by means of a polynomial. This polynomial is called as the chromatic polynomial of G and is defined as follows:

The value of the chromatic polynomial $P_n(\lambda)$ of a graph with n vertices gives the number of ways of properly coloring G using λ or fewer colors.

$P_n(\boldsymbol{\lambda}) = The\ number\ of\ ways\ of\ proper\ coloring\ of\ a\ graph\ with\ n\ vertices, using\ \boldsymbol{\lambda}$ colors

Let c_i be the different ways of properly coloring graph G using exactly i colors.

Since i colors can be choosen out of λ colors in $\binom{\lambda}{i}$ different ways

No. of ways of proper coloring of G exactly i colors out of $\lambda = c_i \binom{\lambda}{i}$

Since i can be any positive integer from 1 to n (it is not possible to use more than n colors on n vertices), the chromatic number is the sum of these terms, that is

$$P_n(\lambda) = \sum_{i=1}^{n} c_i \binom{\lambda}{i}$$

$$= c_1 \frac{\lambda}{1!} + c_2 \frac{\lambda(\lambda - 1)}{2!} + c_2 \frac{\lambda(\lambda - 1)(\lambda - 2)}{3!} + \cdots$$

$$+ c_n \frac{\lambda(\lambda - 1)(\lambda - 2) \ldots (\lambda - n + 1)}{n!}$$

Each c_i has to be evaluated individually for the given graph.

For example, any graph with one edge requires atleast two colors for proper coloring and therefore

$$c_1 = 0.$$

A graph with n vertices and using n different colors can be properly colored in $n!$ ways, i.e.,

$$c_n = n!$$

Chromatic Polynomial of a complete graph:

Number of ways of colorings properly of a complete graph with n vertices and λ colors is called a chromatic polynomial and it is denoted by

$P_n(\lambda) = \lambda(\lambda - 1)(\lambda - 2)\ldots\ldots\ldots(\lambda - (n-1))$

Properties of chromatic polynomial:

1. Degree of $P_n(\lambda)$ is n i.e. equal to the number of vertices.

2. Coefficient of λ^n is one i.e. chromatic polynomial is monic.

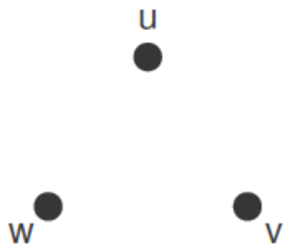
3. The constant term of $P_n(\lambda)$ is zero.

4. The coefficient of $|\lambda^{n-1}|$ is equal to the number of edges in G.

5. The coefficient of λ in disconnected graph is always zero.

5. If the graph is connected the coefficient of λ is nonzero.

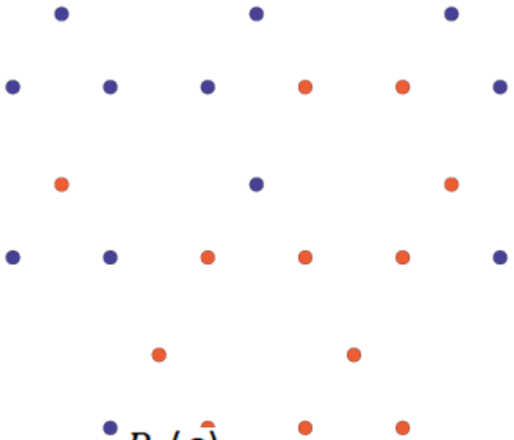
3 Vertices and 0 Edges



⋆ **λ** ways to color independently each of the vertices u, v, w.

$$P_3(\boldsymbol{\lambda}) = \boldsymbol{\lambda}^3$$

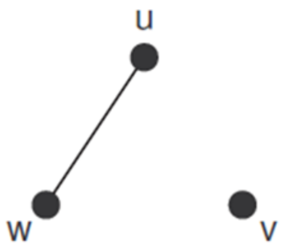
3 Vertices, 0 Edges, and 2 colors



$P_3(2)$

$$P_G(2) = 2^3 = 8$$

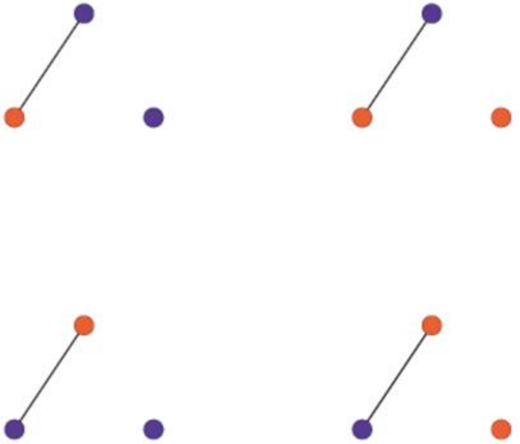
3 Vertices and 1 Edge



⋆ $\boldsymbol{\lambda}$ ways to color v; $\boldsymbol{\lambda}$ ways to color u; $\boldsymbol{\lambda} - 1$ ways to color w that cannot get the color of u.

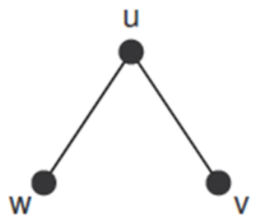
$$
\begin{aligned}
P_3(\boldsymbol{\lambda}) &= \boldsymbol{\lambda}^2(\boldsymbol{\lambda} - 1) \\
&= \boldsymbol{\lambda}^3 - \boldsymbol{\lambda}^2
\end{aligned}
$$

3 Vertices, 1 Edge, and 2 colors



$$P_3(2) = 2^3 - 2^2 = 4$$

3 Vertices and 2 Edges



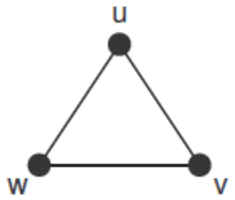
λ ways to color u; $\lambda - 1$ ways to color v that cannot get the color of u; $\lambda - 1$ ways to color w that cannot get the color of u.

$$
\begin{aligned}
P_3(\lambda) &= \lambda(\lambda - 1)^2 \\
&= \lambda^3 - 2\lambda^2 + \lambda
\end{aligned}
$$

3 Vertices, 2 Edges, and 2 colors

$$P_3(2) = 2^3 - 2 \cdot 2^2 + 2 = 2$$

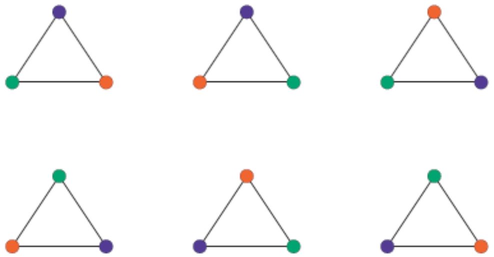
3 Vertices and 3 Edges



$\boldsymbol{\lambda}$ ways to color u; $\boldsymbol{\lambda} - 1$ ways to color v that cannot get the color of u; $\boldsymbol{\lambda} - 2$ ways to color w that cannot get the colors of u and v.

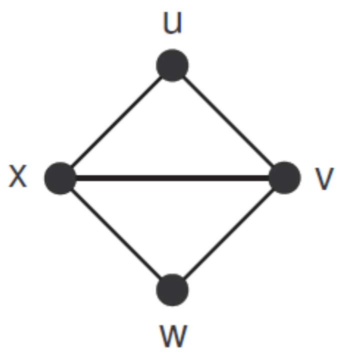
$$\begin{aligned} P_3(\boldsymbol{\lambda}) &= \boldsymbol{\lambda}(\boldsymbol{\lambda} - 1)(\boldsymbol{\lambda} - 2) \\ &= \boldsymbol{\lambda}^3 - 3\boldsymbol{\lambda}^2 + 2\boldsymbol{\lambda} \\ &= (\boldsymbol{\lambda} - 1)^3 - (\boldsymbol{\lambda} - 1) \end{aligned}$$

3 Vertices, 3 Edges, and 3 colors



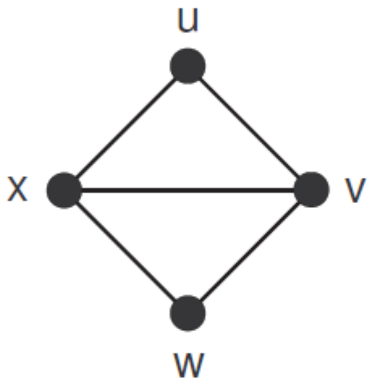
$P_3(3)=\ 3 \cdot 2 \cdot 1 = 6$

4 Vertices and 5 Edges



- ⋆ $\boldsymbol{\lambda}$ ways to color v; $\boldsymbol{\lambda}-1$ ways to color x that cannot get the color of v; $\boldsymbol{\lambda}-2$ ways to color u that cannot get the colors of v and x; $\boldsymbol{\lambda}-2$ ways to color w that cannot get the colors of v and x.

4 Vertices and 5 Edges



$$P_4(\boldsymbol{\lambda}) = \boldsymbol{\lambda}(\boldsymbol{\lambda}-1)(\boldsymbol{\lambda}-2)^2$$
$$= \boldsymbol{\lambda}^4 - 5\boldsymbol{\lambda}^3 + 8\boldsymbol{\lambda}^2 - 4\boldsymbol{\lambda}$$

Null Graphs – N_n

- ⋆ The null graph N_n has n vertices and no edges.
- ⋆ Each vertex can be colored independently with k colors.

$$
\begin{aligned}
P_n\left(\boldsymbol{\lambda}\right) &= \boldsymbol{\lambda}^n \\
&= \boldsymbol{\lambda}^n - 0 \cdot \boldsymbol{\lambda}^{n-1} + 0 \cdot \boldsymbol{\lambda}^{n-2} - \cdots +- 0
\end{aligned}
$$

Stars – S_n

- ⋆ The Star graph S_n has $n-1$ edges. A root vertex is connected to the rest of the $n-1$ vertices each connected only to the root.
- ⋆ The root can be colored with $\boldsymbol{\lambda}$ colors and each of the other $n-1$ vertices can be colored with $\boldsymbol{\lambda}-1$ colors.

$$\begin{aligned} P_n(\boldsymbol{\lambda}) &= \boldsymbol{\lambda}(\boldsymbol{\lambda}-1)^{n-1} \\ &= \boldsymbol{\lambda}^n - (n-1)\boldsymbol{\lambda}^{n-1} + \cdots +- 0 \end{aligned}$$

Theorem: A graph of n vertices is complete graph if and only if its chromatic polynomial is

$$P_n(\boldsymbol{\lambda}) = \boldsymbol{\lambda}\,(\boldsymbol{\lambda}-1)(\boldsymbol{\lambda}-2)\ldots\ldots\ldots(\boldsymbol{\lambda}-(n-1))$$

Proof: Let given graph is complete graph.
Choose any vertex and paint with colors. The number of ways of different coloring for the selected vertex is **λ**.
Now choose second vertex which is adjacent to first vertex and it can be properly colored in (**λ** -1) ways.
Same way the n^{th} vertex will be coloured in (**λ** –(n-1)) ways if and only if every vertex is adjacent to every other. That is if graph is complete.

And chromatic polynomial is

$$P_n(\boldsymbol{\lambda}) = \boldsymbol{\lambda}\,(\boldsymbol{\lambda}-1)(\boldsymbol{\lambda}-2)\ldots\ldots\ldots(\boldsymbol{\lambda}-(n-1))$$

Theorem: An n-vertex graph is a tree iff its chromatic polynomial

$$\mathbf{P_n(\lambda)= \lambda\,(\lambda - 1)^{n-1}}$$

Proof: Let G be a tree

To prove $P_n(\boldsymbol{\lambda})= \boldsymbol{\lambda}\,(\boldsymbol{\lambda} - 1)^{n-1}$

Using induction on n,

For $n = 1$, the result is trivial

i.e. $P_1(\boldsymbol{\lambda})= \boldsymbol{\lambda}$

Assume the result is true for a tree with with atmost $(n-1)$ vertices, where $n \geq 2$.

i.e. $P_{n-1}(\boldsymbol{\lambda})= \boldsymbol{\lambda}\,(\boldsymbol{\lambda} - 1)^{n-2}$.

Let G be a tree with n vertices and e be the pendant edge of G.

Now $P_n(\boldsymbol{\lambda})= P_{n-1}(\boldsymbol{\lambda})P_1(\boldsymbol{\lambda})$

$=\boldsymbol{\lambda}\,(\boldsymbol{\lambda} - 1)^{n-1}$.

Conversely, Let G be a simple graph with chromatic polynomial

$\mathbf{P_n(\lambda)= \lambda\,(\lambda - 1)^{n-1}}$

$= \boldsymbol{\lambda}(\boldsymbol{\lambda}^{n-1} - (n-1)\boldsymbol{\lambda}^{n-2} + \ldots\ldots\ldots + (-1)^{n-1}$

$= \boldsymbol{\lambda}^{n} - (n-1)\boldsymbol{\lambda}^{n-1} + \ldots\ldots\ldots + (-1)^{n}\,\boldsymbol{\lambda}$

which is monic polynomial with n vertices.

The highest degree in n.

Also the last term $(-1)^{n}\,\boldsymbol{\lambda}$ ensures that G is connected.

The next highest degree is $(n-1)$ and coefficient of $|\boldsymbol{\lambda}^{n-1}|$ is $(n-1)$, which gives number of edges in the graph.

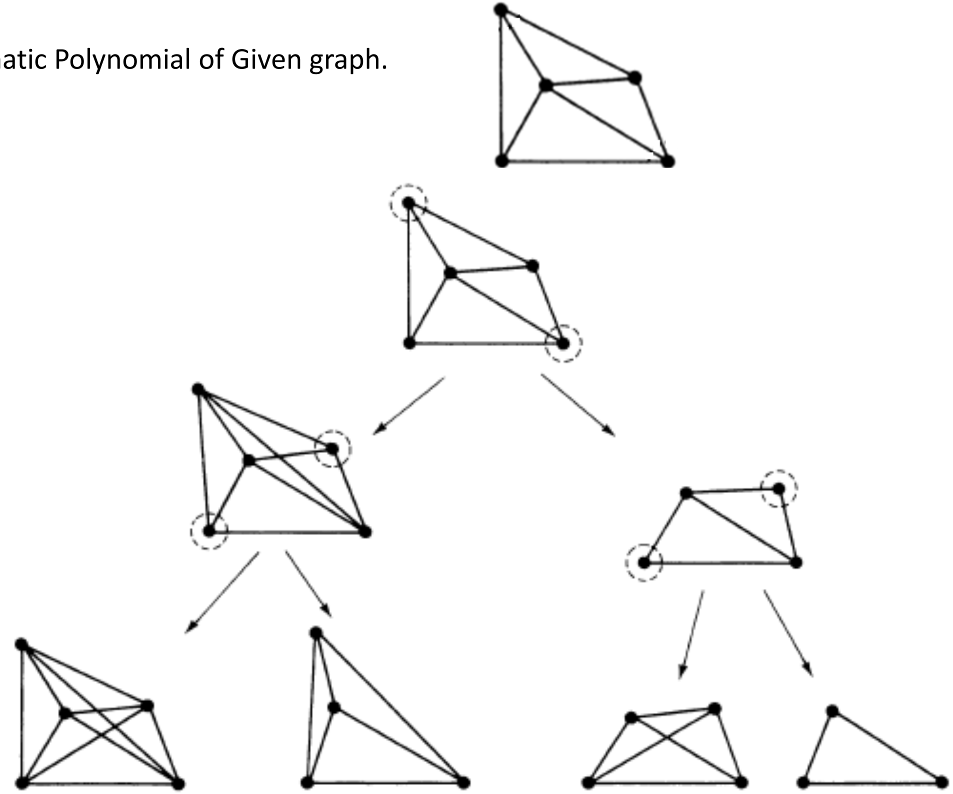
i.e. G is a Tree.

THEOREM

Let a and b be two nonadjacent vertices in a graph G. Let G' be a graph obtained by adding an edge between a and b. Let G'' be a simple graph obtained from G by fusing the vertices a and b together and replacing sets of parallel edges with single edges. Then

$$P_n(\lambda) \text{ of } G = P_n(\lambda) \text{ of } G' + P_{n-1}(\lambda) \text{ of } G''.$$

Ex: Find Chromatic Polynomial of Given graph.



$\lambda(\lambda - 1)(\lambda - 2)(\lambda - 3)(\lambda - 4)$ $\quad$ $\lambda(\lambda - 1)(\lambda - 2)(\lambda - 3)$ $\quad$ $\lambda(\lambda - 1)(\lambda - 2)(\lambda - 3)$ $\quad$ $\lambda(\lambda - 1)(\lambda - 2)$

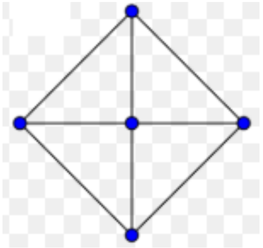
$$
\begin{aligned}
P_5(\lambda) \text{ of } G &= \lambda(\lambda-1)(\lambda-2) + 2\lambda(\lambda-1)(\lambda-2)(\lambda-3) \\
&\quad + \lambda(\lambda-1)(\lambda-2)(\lambda-3)(\lambda-4) \\
&= \lambda(\lambda-1)(\lambda-2)(\lambda^2-5\lambda+7)
\end{aligned}
$$

Evaluation of a chromatic polynomial.

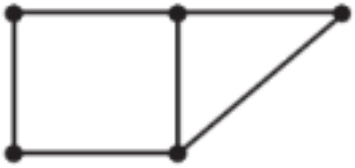
Ex: Find Chromatic Polynomial of Given graph.

1.

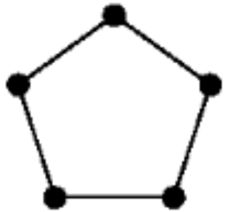
2.



3.



4.



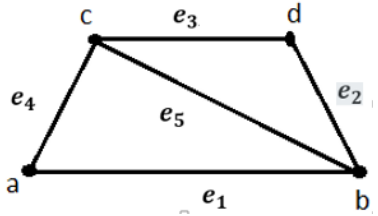
Matching:

Matching in a graph is subset of edges in which no two edges are adjacent.

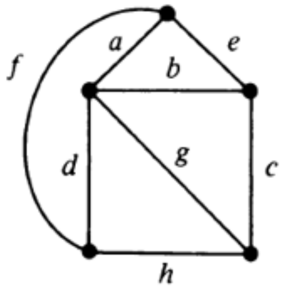
Remark: A single edge in a graph is obviously a matching.

Example:
Matching are $\{\boldsymbol{e_1}\}$, $\{\boldsymbol{e_2}\}$, $\{\boldsymbol{e_3}\}$, $\{\boldsymbol{e_4}\}$,$\{\boldsymbol{e_5}\}$,

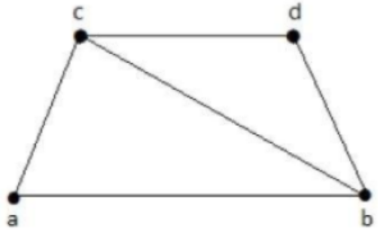
$\{\boldsymbol{e_1}, \boldsymbol{e_3}\}, \{\boldsymbol{e_2}, \boldsymbol{e_3}\}$

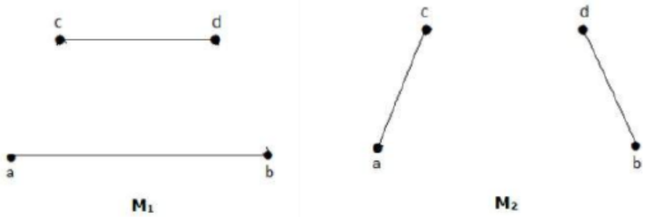


Example: In the given graph the matchings are
M_1=$\{h, b\}, M_2 = \{a, c\quad\}, M_3 = \{e, g\qquad\}, M_4 = \{c, d\}, M_5 = \{a, c\}$ etc .



Example:

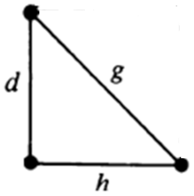




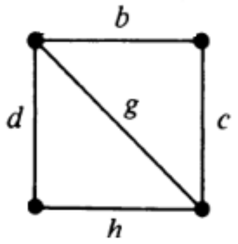
Maximal Matching:

A matching of the graph is said to be maximal in which no other edges can be added.

Ex1 : In a complete graph of three vertices any single edge is a matching.
i.e. maximal matching are $\{d\}, \{h\}, \{g\}$.



Ex 2 : In the given graph the maximal matching are $\{b, h\}, \{d, c\}, \{g\}$ is the maximal matching.

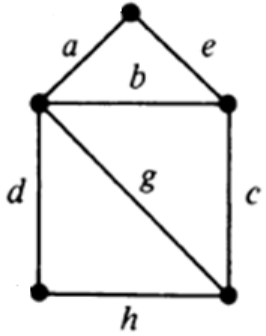


Ex 3 : Graph and its two maximal matching are

(a) (b)

Graph and two of its maximal matchings.

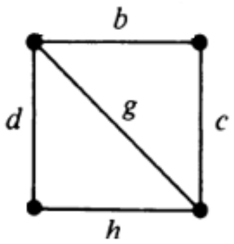
Ex: $\{h, b\}, \{d, c\}, \{d, e\}, \{c, a\}, \{g, e\}, \{h, a\}, \{h, e\}$ are maximal matchings of given graph.



Largest Maximal Matching or Maximum Matching:

A maximal matching of the graph with largest number of edges is defined as largest matching.

Ex: In the given graph the largest matchings are $\{b, h\}, \{d, c\}$.



Matching Number:

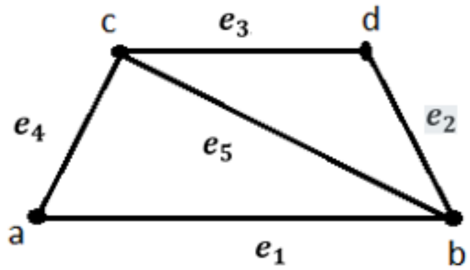
The number of edges in largest matching of G is called matching number.

Ex: In above example the matching number is 2.

Perfect Matching or Complete Matching:

A matching M of a graph G is said to be a perfect match if every vertex of G is incident to exactly one edge of the matching M.

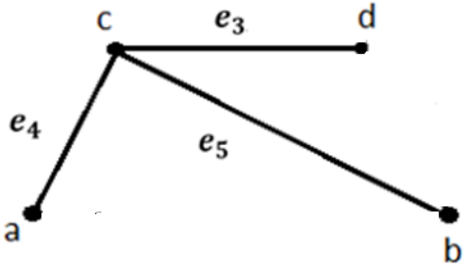
Example: $M_1 = \{e_1,\ e_3\}, M_2 = \{e_2, e_4\}$ are are perfect matching of G.



Note: Every perfect matching of graph is also a maximum matching of a graph, because there is no chance of adding one more edge in a perfect matching graph. But a maximum matching need not to be a perfect matching.

Note: If a graph G has perfect matching then number of vertices is always even but the converse need not be true.

$M_1 = \{e_4\}, M_2$=$\{e_5\}$,$M_3 = \{e_3\}$ are maximal matching of given graph but not perfect matching Though it has even number of vertices.



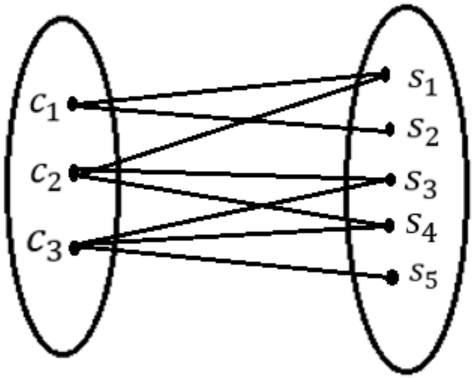
Note: If the number of vertices in the graph is odd then perfect matching does not exist but the converse need not to be true.

Complete matching in a Bipartite graph:

Theorem: A complete matching of V_1 into V_2 in a bipartite graph exist if and only if every subset of r vertices in V_1 is collectively adjacent to r or more vertices V_2 for all values of r.

Example:

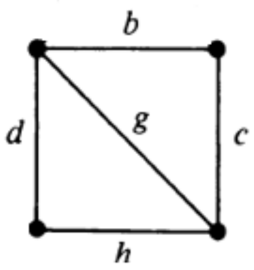
Complete matching in bipartite graph.



Covering

Edge Covering or line Covering:

In a graph G a set of edges g is said to be cover G if every vertex in G is incident on at least one edge in g.
A set of edges that covers a graph G is said to be an edge covering or cove



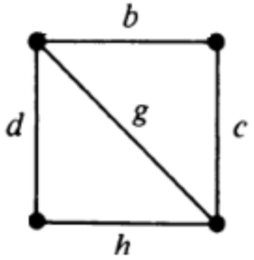
Ex: A graph G is trivially its own covering.
A spanning tree in a connected graph is another covering.
A Hamiltonian circuit in a graph is also a covering(if it exists).

Ex: Edge covering are $\{b, g, h\}$, $\{d, c\}$,$\{b, h\}$, $\{b, d, h\}$, $\{b, c, h\}$, $\{d, b, c\}$ etc

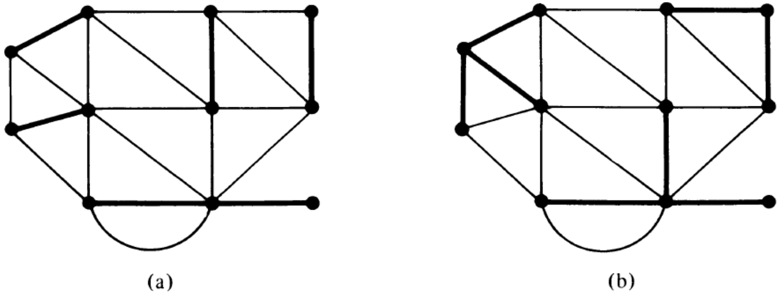
Minimal edge covering:

An **edge covering** of a graph G is said to be minimal if no edge can be removed without destroying its ability to cover the graph.

Ex:$\{b, h\}, \{d, c\}, \{b, d, g\}$ are minimal line covering of G.



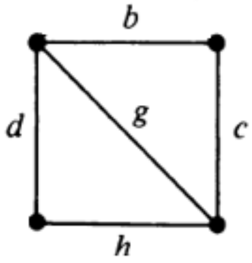
Ex:



Graph and two of its minimal coverings.

Minimum Edge covering:

A minimal line covering with minimum number of edges is called minimum line covering.
Ex: :$\{b, h\}, \{d, c\}$



Edge covering number:

A graph , in general has many edge covering , and they may be of different sizes.(consisting of different number of edges)

The number of edges in a minimum line covering in G is called the line covering number of G.
covering number =2 for given graph G.

- A covering exists for a graph if and only if the graph has no isolated vertex.
- A covering of n-vertex graph will have at least $\lceil n/2 \rceil$ edges. ($\lceil x \rceil$ denotes the smallest integer not less than x) .
- Every line covering contains a minimal covering.
- No minimal covering can contain a circuit for we can always remove an edge from a circuit without leaving any of vertices in the circuit uncovered. Therefore, a minimal covering of an n-vertex graph can contain no more than $n - 1$ edges.

Theorem:

A covering $\boldsymbol{g}$ of a graph G is minimal if and only if $\boldsymbol{g}$ contains no path of length three or more.

Proof: Let $\boldsymbol{g}$ be a minimal covering of a graph and we will prove that $\boldsymbol{g}$ contains no path of length three or more.

Let $\boldsymbol{g}$ contains a path of length three and it is $V_1\ e_1\ V_2\ e_2\ V_3\ e_3\ V_4$.

Edge e_2 can be removed without leaving its end vertices v_2 and v_3 uncovered, still $\boldsymbol{g}$ is a covering, which is a contradiction that $\boldsymbol{g}$ is a minimal covering. Hence $\boldsymbol{g}$ contains no path of length three or more.

Conversely, let a covering $\boldsymbol{g}$ contains no path of length three or more, therefore all its component must be a star graph (i.e., graphs in the shape of stars). From a star graph no edge can be removed without leaving a vertex uncovered, hence, $\boldsymbol{g}$ must be a minimal covering.

Four color problem

So far we have considered proper coloring of vertices and proper coloring of edges. Now, let us consider the proper coloring of regions in a planar graph.

The regions of the planar graph are said to be properly colored if no two contiguous or adjacent regions have the same color. (Two regions are said to be adjacent if they have a common edge between them.)

The proper coloring of the regions is also called map coloring.

Example:

In atlas different countries are colored such that countries with common boundaries are shown in different colors.

Four color conjecture:

To color the regions of the graph with the minimum number of colors leads us to the most famous conjecture in graph theory which stated that every map (i.e., a planar graph) can be properly colored with four colors.