CHAPTER 04

Graph Theory

4.1 Fundamental Concepts

Graph: A graph G is a triple consisting of a vertex set V(G), an edge set E(G), and a relation that associates with each edge two vertices called its end points. Vertices are sometimes called as nodes.

A loop is an edge whose end points are equal. Multiple edges are edges having the same pair of end points.

4.1.1 Types of Graphs

Simple Graph

A Simple graph is an undirected graph having no loops or multiple edges. We specify a simple graph by its vertex set and edge set treating the edge set as a set of unordered pairs of vertices (undirected graph) and writing $e = \{u, v\}$ for an edge e with endpoints u and v.

When u and v are the endpoints of an edge, they are adjacent and are neighbors. We write $u \leftrightarrow v$ for u is adjacent to v and we say that the edge

$e = \{u, v\}$ is incident on u and v.

Multigraph

A Multigraph is an undirected graph in which multiple edges between pairs of vertices is allowed. However, self loops are not allowed.

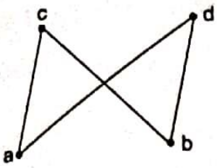
Psuedograph

A psuedograph is an undirected graph in which multiple edges as well as self loops are allowed.

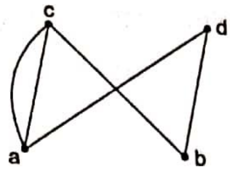
4.1.2 Directed Graph

A graph in which the edges have direction. i.e. A graph (V, E) such that E is a set of directed edges that are ordered pairs of vertices of V.

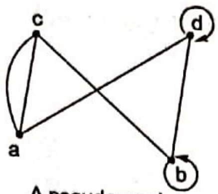
NOTE: In an undirected graph, E is a set of undirected edges that are unordered pairs of vertices of V.



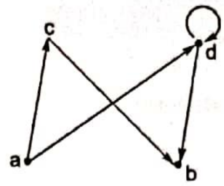
A simple graph



A multigraph



A pseudograph



A digraph

Vertex Degrees & Counting

The **degree** of a vertex v in a graph G, written $d_G(v)$ or $d(v)$, is the number of edges, incident to v, except that each loop at v counts twice. The maximum degree is $\Delta(G)$, the minimum degree is $\delta(G)$, and G is regular if $\Delta(G) = \delta(G)$. i.e. all vertices have the same degree. It is **k-regular** if the common degree is k. The neighbourhood of v written $N(v)$, is the set of vertices adjacent to v.

The **order** of a graph G, written $n(G)$, is the number of vertices. An "n-vertex graph" is a graph of order n. The **size** of a graph G written $e(G)$, is the number of edges in G.

Degree-sum formula for undirected graphs (Handshaking theorem): If G is an undirected graph, then

$$\sum_{v \in V} d(v) = 2e.$$

Corollary

In a graph G, the average vertex degree is $\frac{2e}{n}$ and $\delta(G) \leq \frac{2e}{n} < \Delta(G)$.

NOTE: A vertex with zero degree is called a **lone vertex** or isolated vertex and a vertex with exactly one degree is called a **pendent** vertex or end vertex.

Some Results:

1. Every graph has an even number of vertices of odd degree. No graph of odd order is regular with odd degree.
2. A k-regular graph with n vertices has $\frac{nk}{2}$ edges.
3. The minimum number of edges in a connected graph with n vertices is $(n-1)$.
4. If G is a simple n-vertex graph with $\delta(G) \geq \frac{(n-1)}{2}$ then G is connected.

4.1.3 Directed Graphs

In a directed graph, if (u, v) is an edge in G, then u is **adjacent** to v and v is adjacent to u. u is called **initial vertex** and v is called **terminal vertex** of the edge (u, v).

Indegree and outdegree of a vertex (v) in a directed graph: The indegree of a vertex v denoted by $d^+(v)$ is the number of edges with v as their terminal vertex. The **out degree** of a vertex denoted by $d^-(v)$ is the no of edges with v is their initial vertex.

Loops contribute 1 to the indegree and 1 to out degree of a vertex on which they are present.

Degree-sum Formula for Directed Graphs

$$\Sigma d^-(v) = \Sigma d^+(v) = e$$

i.e. $$\Sigma d^-(v) + \Sigma d^+(v) = 2e$$

Example-4.1 Given a graph with 10 vertices and 12 edges. What is highest value of minimum degree?

Solution:

$$\delta_{max} = \left\lfloor \frac{2e}{n} \right\rfloor = 2$$

is the highest value of minimum degree that is possible.

Also the least value of maximum degree is

$$\left\lceil \frac{2e}{n} \right\rceil \leq \Delta$$

i.e.

$$\left\lceil \frac{2 \times 12}{10} \right\rceil \leq \Delta$$

$$3 \leq \Delta$$

Hence the least value of maximum degree that is possible is 3.

Example-4.2 Given a tree with n_2 vertices with degree 2, n_3 vertices with degree 3, n_4 vertices with degree 4, and so on such that n_k vertices with degree k. How many leaf nodes are there?

Solution:

$$\text{Total degree} = x * 1 + 2 * n_2 + 3 * n_3 + \ldots + k * n_k$$

In tree the number of edges $(|E|) = |V| - 1$

i.e. $x + 2n_2 + 3n_3 + \ldots + kn_k = 2(x + n_1 + n_2 + \ldots + n_k - 1)$

Solving for x, we get

$$x = n_3 + 2n_4 + 3n_5 + \ldots + (k-2)\,n_k + 2$$

'x' is the number of vertices with degree 1 and so are the leaf nodes.

Example-4.3 What is the order and size of $\bar{G}$, given that order and size of G are 5 and 7 respectively?

Solution:

$$O(\bar{G}) = 5$$

Because the number of vertices do not change in the compliment of a graph.

Now,

$$\begin{aligned} \text{Size} &= \text{Number of edges} \\ &= \text{Number of edges in } k_5 - \text{Number of edges in G} \\ &= \frac{5(5-1)}{2} - 7 = 3 \end{aligned}$$

4.1.4 Havell-Hakimi Theorem

Let G be a graph with vertex set

$$V(G) = \{V_1, V_2, V_3 \ldots V_n\}$$

and let deg V_1, deg V_2, ..., deg V_n be the degree sequence.

NOTE: A sequence $d_1, d_2, \ldots, d_n$ of non-negative integer is graphical if it is a degree sequence of some graph. We study a powerful tool to determine whether a particular sequence is graphical.

Theorem

Let D be the sequence $d_1, d_2, ...d_n$ with $d_1 \geq d_2 \geq ... \geq d_n$ and $n \geq 2$.
Let D' be the sequence obtained from D by

→ discarding d_1 and

→ subtracting 1 from each of the next d_1 entries of D

i.e. D' is the sequence.

$d_2-1, d_3-1, d_4-1,...., d_{d_1+1}-1, d_{d_1+2}-1,...,d_n$ when sequence has only 1's and 0's and the number of 1's are even then we say that the sequence is graphical.

Example-4.4 Given a graph with 14 edges. What is the missing value 'k' in the following degree sequence, 1, 2, 3, 3, 4, k, k, k.

Solution:

By using hand shaking theorem,

$$2 \times 14 = 1 + 2 + 3 + 3 + 4 + 3k$$
$$28 = 13 + 3k$$
$$15 = 3k$$
$$k = 5$$

Hence the actual degree sequence is 1, 2, 3, 3, 4, 5, 5, 5.

Example-4.5 Check whether the following sequences are valid degree sequences or not

(a) 5, 3, 3, 3, 2, 2, 1, 1 (b) 6, 5, 5, 5, 4, 4, 2, 1 (c) 8, 7, 6, 6, 5, 3, 2, 2, 2, 1

Solution:

(a) 5, 3, 3, 3, 2, 2, 1, 1

Apply Havel-Hakimi theorem,

5, 3, 3, 3, 2, 2, 1, 1 is graphical iff

2, 2, 2, 1, 1, 1, 1 is graphical iff {obtained by subtracting 1 from next 5 degree in descending order}

1, 1, 1, 1, 1, 1 is graphical {obtained by subtracting 1 from next 2 degree in descending order}

The last sequence is graphical hence all the above sequence would be graphical.

Note: Here the last sequence 1, 1, 1, 1, 1, 1 is graphical only because number of 1's are even.

(b) 6, 5, 5, 5, 4, 4, 2, 1

4, 4, 4, 3, 3, 1, 1

3, 3, 2, 2, 1, 1

2, 1, 1, 1, 1

0, 0, 1, 1

Here number of 1's are even graphical sequence.

(c) 8, 7, 6, 6, 5, 3, 2, 2, 2, 1

6, 5, 5, 4, 2, 1, 1, 1, 1

4, 4, 3, 1, 0, 0, 1, 1

Rearrange in decreasing order

4, 4, 3, 1, 1, 1, 0, 0

3, 2, 0, 0, 1, 0, 0

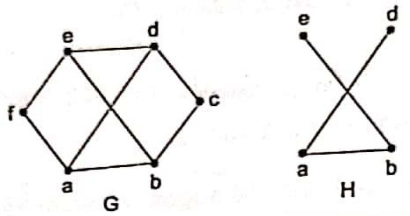
1, 0, −1, 0, 0, 0, 0

Hence not a graphical sequence

4.1.5 Subgraphs

A subgraph of a graph G is a graph H such that $V(H) \subseteq V(G)$ and $E(H) \subseteq E(G)$ and the assignment of end points to edges in H is the same as in G. We then write $H \subseteq G$ and say that "G" contains H"

Example: Here $H \subseteq G$



NOTE: Every graph is its own subgraph. $G \subseteq G$. Null graph is a subgraph of every graph.

4.2 Special Graphs

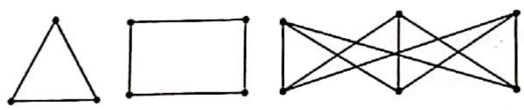
(a) Finite Graph: A graphs is finite if its vertex set and edge set are both finite. If a graph is not finite, it is infinite.

(b) Null Graph: A graph with vertices but no edges. i.e. A graph in which the edge set is empty, but vertex set is not empty is called a null graph.

NOTE: The vertex set cannot be empty, i.e. a graph must have at least one vertex.

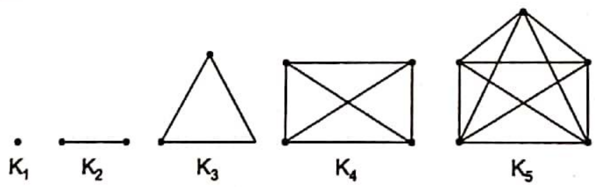
(c) Regular Graph: A graph is which all vertices have the same degree is called a regular graph. In a regular graph $\delta(G) = \Delta F$

Examples:



NOTE: A regular graph with $\delta(G) = \Delta G = k$ is called k-regular.

(d) Complete Graph: The complete graph of n vertices denoted by K_n is a simple graph that contains exactly one edge between every pair of distinct vertices.

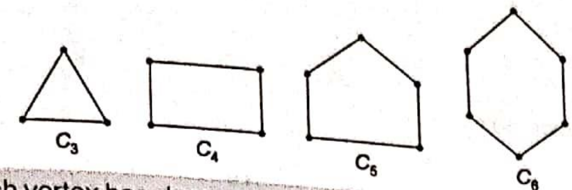


NOTE

- A complete graph is a simple graph with maximum number of possible edges.
- The number of edges in $K_n = \dfrac{n(n-1)}{2} = nC_2$

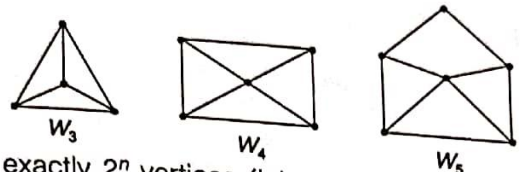
  Example: K_5 has 5 vertices and $\dfrac{5(5-1)}{2} = 10$ edges
- If a simple graph has nC_2 edges it is complete.
- Maximum edges possible in an n-vertex simple graph is $\dfrac{n(n-1)}{2}$

(e) Cycle Graph: The cycle C_n, $n \geq 3$ consists of n vertices $v_1, v_2, \ldots, v_n$ and edges $\{v_1, v_2\}, \{v_2, v_3\}\ldots$ and so on until $\{v_n, v_1\}$.

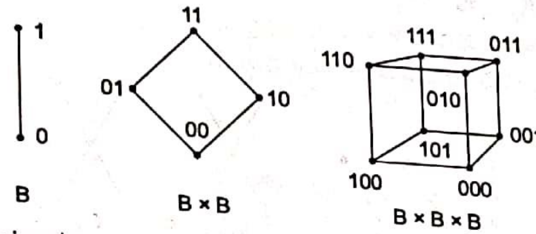


NOTE: In a cycle graph each vertex has degree 2 and all cycle graphs are 2-regular.

(f) Wheel Graph: A cycle graph C_n with a vertex v_{n+1} added in centre and edges $\{v_1, v_{n+1}\}, \{v_2, v_{n+1}\}\ldots \{v_n, v_{n+1}\}$ also added, makes a wheel graph W_n.

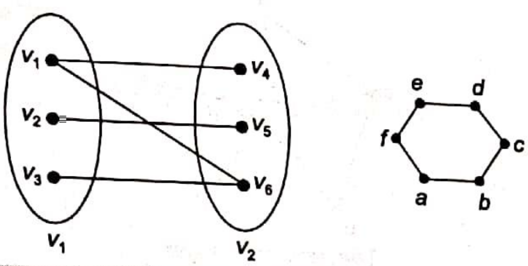


(g) n-Cubes: Consist of exactly 2^n vertices (labelled by bit strings) and connected by the relation 2 bitstrings are adjacent iff Hamming distance between them is 1.



(h) Bipartitc Graphs: A simple graph G is called bipartite if its vertex set V can be partitioned into two distinct non empty sets, V_1, V_2 such that any edge in G connects a vertex in V_1 with a vertex in V_2 (so that no edge connects either 2 vertices in V_1 or 2 vertices in V_2).

Example:

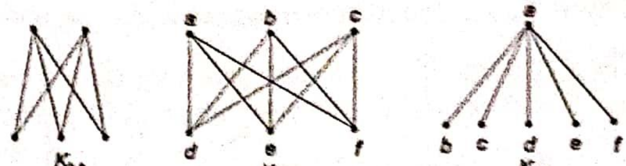


NOTE: Cycle graph C_6 above is bipartite $V_1 = \{a, c, e\}$ and $V_2 = \{b, d, f\}$. K_4, K_3 are not bipartite, but K_2 is bipartite.

Lemma: A graph is bipartite iff it has no odd cycle. We can show a graph to be not bipartite, if we detect an odd cycle (Biclique).

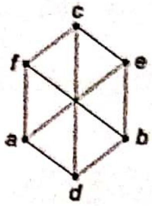
(i) Complete Bi-partite Graphs: $K_{m,n}$ is a graph that has its vertex set partitioned into two subsets V_1 and V_2 of m and n vertices, respectively. There is an edge between 2 vertices iff one vertex is in first subset and another vertex is in second subset.

In other words, in a complete bipartite graph every vertex in v_1 is connected to every vertex v_2 by an edge.

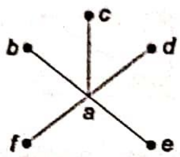


$K_{2,3}$ $K_{3,3}$ $K_{1,5}$

$K_{3,3}$ can also be drawn as:



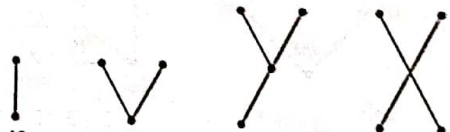
$K_{1,5}$ can also drawn as:



Result:

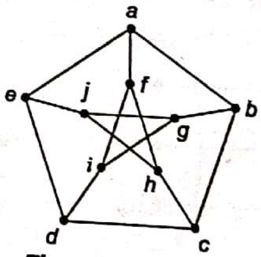
(*i*) The no of vertices in $K_{m,n}$ is exactly $m + n$ and the no of edges in $K_{m,n}$ is exactly mn. i.e. $K_{3,4}$ has exactly $3 + 4 = 7$ vertices and $3 \times 4 = 12$ edges.

(*ii*) Any graph that is $K_{1,n}$ is also called a star graph.



$K_{1,1}$ $K_{1,2}$ $K_{1,3}$ $K_{1,4}$

(j) **Petersen Graph:** The petersen graph is the simple graph whose vertices are the 2-element subsets of a 5-element set and whose edges are the pairs of disjoint 2-element subsets. If two vertices are non adjacent in the petersen graph, then they have exactly one common neighbor.



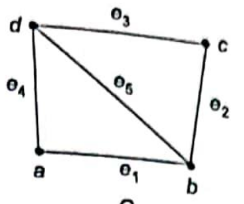
The petersen graph

The **girth** of a graph with a cycle is the length of its shortest cycle. A graph with no cycle has infinite girth. The Petersen graph has girth 5.

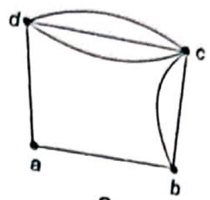
4.3 Graph Representations

Definition: Let G be a loopless graph with vertex set V(G) = $\{v_1, \dots v_n\}$ and edge set E(G) = $\{e_1, \dots e_n\}$. The **adjacency matrix** of G, written A(G), is the n-by-n matrix in which entry $a_{i,j}$ is the number edges in G with end points $\{v_i, v_j\}$. The **incidence matrix** M(G) is the n-by-n matrix in which entry $m_{i,j}$ is 1 if v_i is an end point of e_j. If vertex v is an endpoint of edge e, the v and e are **incident**. The **degree** of vertex v is the number of edges, incident upon it.

Example:



G_1



G_2

Matrix representations:

Adjacency matrix of G_1 is given below:

$$X_1 = \begin{array}{c} \\ a \\ b \\ c \\ d \end{array} \begin{array}{c} \begin{array}{cccc} a & b & c & d \end{array} \\ \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix} \end{array}$$

Adjacency matrix of G_2

$$X_2 = \begin{array}{c} \\ a \\ b \\ c \\ d \end{array} \begin{array}{c} \begin{array}{cccc} a & b & c & d \end{array} \\ \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 2 & 1 \\ 0 & 2 & 0 & 3 \\ 1 & 0 & 3 & 0 \end{bmatrix} \end{array}$$

Incidence matrix of G_1

$$A = \begin{array}{c} \\ a \\ b \\ c \\ d \end{array} \begin{array}{c} \begin{array}{ccccc} e_1 & e_2 & e_3 & e_4 & e_5 \end{array} \\ \begin{bmatrix} 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 \end{bmatrix} \end{array}$$

Adjacency list representation:

The Adjacency list for G_1 is

Vertex	Adjacent Vertices
a	b, d
b	a, c, d
c	b, d
d	a, b, c

4.4 Isomorphism

An **isomorphism** from a simple graph g to a simple graph H is a bijection $f: V(G) \to V(H)$ such that $uv \in (G)$ iff $f(u)\, f(v) \in E(H)$.

We say "G is isomorphic to H" written as $G \equiv H$, if there is an isomorphsion from G to H.

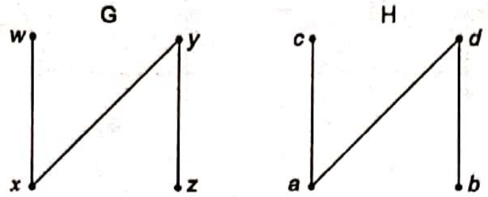
Example: The graph G and H drawn below are 4-vertex paths.

Define the function $f: V(G) \to V(H)$ by $f(w) = a$, $f(x) = d$, $f(y) = b$, $f(z) = c$.

To show that f is an isomorphism, we check that f preserves edges and non-edges.

Note that rewriting A(G) by placing the rows in the order w, y, z, x and the column also in that order yields A(H), this verifies that f is an isomorphism.

Another isomorphism maps w, x, y, z to c, b, d, a respectively.



$$
\begin{array}{c|cccc}
 & w & x & y & z \\
\hline
w & 0 & 1 & 0 & 0 \\
x & 1 & 0 & 1 & 0 \\
y & 0 & 1 & 0 & 1 \\
z & 0 & 0 & 1 & 0
\end{array}
\qquad
\begin{array}{c|cccc}
 & w & y & z & x \\
\hline
w & 0 & 0 & 0 & 1 \\
y & 0 & 0 & 1 & 1 \\
z & 0 & 1 & 0 & 0 \\
x & 1 & 1 & 0 & 0
\end{array}
$$

$$
\begin{array}{c|cccc}
 & a & b & c & d \\
\hline
a & 0 & 0 & 0 & 1 \\
b & 0 & 0 & 1 & 1 \\
c & 0 & 1 & 0 & 0 \\
d & 1 & 1 & 0 & 0
\end{array}
$$

Finding Isomorphisms

Presenting the adjacency matrices, with vertices ordered so that the matrices are identical, is one way to show an isomorphism.

One can also verify preservation of adjacency relation without writing out the matrices.

In order for an explicit bijection to be an isomorphism from G to H, the image in H of a vertex v in G must behave in H as v does in G.

The isomorphism relation is an equivalence relation on the set of (simple) graphs.

An "isomorphism class" of graphs is an equivalence class of graphs under the isomorphism relation.

NOTE: An automorphism of G is an isomorphism from G to G. A graph G is vertex transitive if for every pair $u, v \in V(G)$ there is an automorphism that maps u to v.

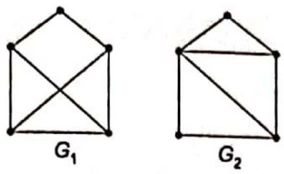
4.5 Invariants of Isomorphic Graphs

Isomorphic simple graphs must have

1. Same number of vertices
2. Same number of edges.
3. Same degree sequence. i.e. degree of corresponding vertices must be same.
4. Number of simple circuits of a given length must be same in both graphs.

All the above are called invariants in an isomorphism. If any of the above are violated, then the graphs are not isomorphic. Note that the above invariants are necessary, but not sufficient to prove isomorphism but violation of these invariants can be used to prove that the graphs are not isomorphic.

Example:



Although G_1 and G_2 have same number of vertices as well as same number of edges, they are not isomorphic since there degree sequences are not the same. The degree sequence of G_1 is (2,3,3,3,3) and the degree sequence of G_2 is (2,2,3,3,4).

4.6 Operations on Graphs

Complement of a Graph

If H is a simple graph with n vertices, the complement $\overline{H}$ of H is the complement of H in K_n i.e. $\overline{H}$ is the graph of $K_n - E(H)$.

From this definition, we can say that,

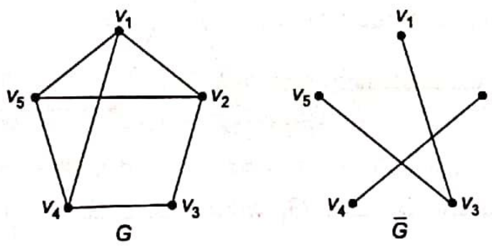
(i) $V(\overline{H}) = V(H)$ and $e(\overline{H}) = \frac{n(n-1)}{2} - e(H)$ (where $n = V(H) = \#$ of vertices of H)

(ii) Any two vertices are adjacent in $\overline{H}$ iff they are not adjacent in H.

(iii) The degree of a vertex in $\overline{H}$ plus its degree in H is $n-1$, when $n = |V(H)|$.

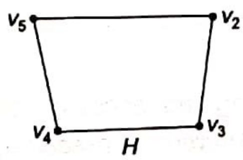
NOTE: If G is a simple graphs of n vertices, then $G \cup \overline{G}$ is K_n, the complete graph on n vertices.

Example:

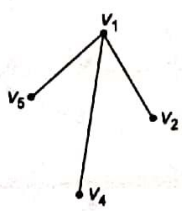


Relative complement of a Graph

Let H be a graph as shown below. $\overline{H}(G) = G - H$, is the relative complement of H in G. It is obtained by starting with G and deleting the edges common to H and G.



Now, $\overline{H}(G) = G - H$ is shown below:



NOTE: Two simple graphs are isomorphic iff their complements are isomorphic.

Decompositions and Special Graphs

A graph is **self-complementary** if it is isomorphic to its complement.

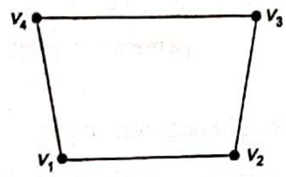
A **decomposition** of a graph is a list of subgraphs such that each edge appears in exactly one subgraph in the list.

An n-vertex graph H is self-complementary if and only if K_n has a decomposition consisting of two copies of H.

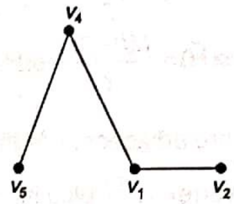
Union and Intersection of Graphs

Let G and G′ be two graphs. The **intersection** of G and G′ written as $G \cap G'$, is the graph where vertex set is $V(G) \cap V(G')$ and where edge set is $E(G) \cap E(G')$. Similarly the **union** of G and G′ is the graph with vertex set $V(G) \cup V(G')$ and edge set $E(G) \cup (G')$.

Example: Let G and G′ be two graphs shown below.

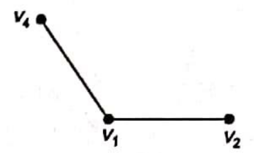


G

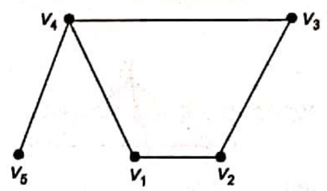


G

Then $G \cap G'$ and $G \cup G'$ are shown below:



$G \cap G'$



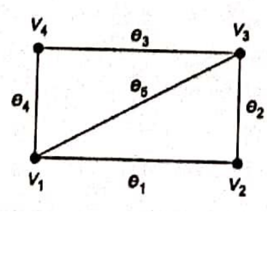
$G \cup G'$

Definition: The "union" of graph G_1, G_2 ...G_k, written $G_1 \cup G_2 \cup \ldots \cup G_k$ is the graph with vertex set $\bigcup_{i=1}^{k} V(G_i)$ and edge set $\bigcup_{i=1}^{k} E(G_i)$. The complete graph K_n can be expressed as the union of k bipartite graphs if and only if $n \leq 2^k$.

4.7 Walks, Paths and Cycles

A **walk** is defined as a finite alternating sequences of vertices and edges, beginning and ending with vertices, such that each edge is incident with the vertices preceding and following it. No edge appears more then once in a walk. A vertex may however appear more than once.

Example:



The sequence $v_1e_1v_2e_2v_3e_5v_1e_4v_4$ is a walk. The set vertices and edges constituting a walk is clearly a subgraph of G.

The vertices with which a walk begins and ends are called terminal vertices. If a walk starts and ends at the same vertex, it is called a closed walk. Else it is a open walk. A open walk, is which no vertex appears more than once is called a **path** or a simple path or elementary path. A path is basically a non-self intersecting open walk.

The no of edges is a path is called the length of the path. A self loop can be included in a walk but not in a path. A closed walk in which no vertex (except the initial and final vertex) appears more than once, is called a **circuit or cycle**. A circuit is therefore a closed, non intersecting walk.

In the graph above, $v_1e_1v_2e_2v_3e_5v_1$ is a circuit or a cycle of length 3.

NOTE: A self loop is a cycle of length 1. A walk from *u* to *v* is called a *u-v* walk. Every *u-v* walk contains a *u-v* path also.

4.8 Connected Graphs, Disconnected Graphs and Components

A graph G is said to be **connected** if there is at least one path between every pair of vertices in G. Otherwise, G is **disconnected**.

A disconnected graph consists of two or more connected graphs. Each of these connected subgraphs is called a **component**.

∴ A connected graph is a graph with only one component. A disconnected graph has at least 2 components. A null graph with more than one vertex is disconnected.

Results:

1. If a graph (connected or disconnected) has exactly 2 vertices of odd degree, there must be a path joining these two vertices.
2. A simple graph with n vertices and k components can have at most $(n-k)(n-k+1)/2$ edges.

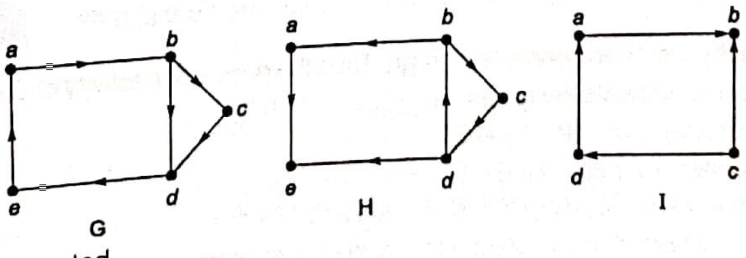
NOTE: From above, when $k=1$, a simple graph with n vertices can have atmost $(n-1)\,n/2$ edges which is the maximum possible edges in a connected simple graph.

3. Every graph with n vertices and k edges has at least $n-k$ components (where $n \geq k$)

Connectedness in Directed Graphs

1. A directed graph is strongly connected, if there is a path from *a* to *b* and from *b* to *a* whenever *a* and *b* are vertices in the graph.
2. A directed graph is weakly connected, if there is a path between every two vertices in the underlying undirected graph.
3. A directed graph is unilaterally connected, if for any pair of vertices *a*, *b*, either there is a path from *a* to *b* or from *b* to *a*.

Example:



G is strongly connected.
H is weakly connected, and I is unilaterally connected, but not strongly connected.

NOTE

- Strongly connected ⇒ Unilateral as well as weakly connected but converse is not true
- Unilaterally connected ⇒ Weakly connected but converse is not true
- If a diagraph is not even weakly connected, then such a graph will be a disconnected graph.

4.8.1 Cut vertex, Cut set and Bridge

A **cut vertex** of a graph G is a vertex whose removal increases the number of components. Clearly, if v is a cut vertex of a connected graph G, then its removal must disconnect the graph. A cut vertex is also called a cut point, a cut node or an articulation point.

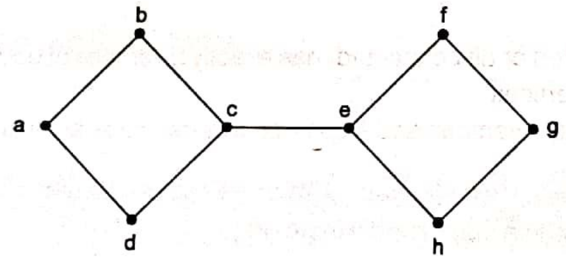
Similarly, an edge whose removal produces a graph with more components than the original graph is called a **cut edge** or a bridge.

Clearly, if e is a cutedge of a connected graph, its removal would surely disconnect the graph.

The set of all minimum number of edges of G whose removal disconnects a graph G is called a **cut set** of G. Thus a cut set S satisfies the following conditions.

1. S is a subset of the edge set E of G.
2. Removal of edges in S from G, disconnects G.
3. No proper subset of S satisfies this condition.

Example:



In above graph, vertex "c" is a cut vertex. So is vertex "e".
Edge {c, e} is a cut edge.
{{c, e}} is a cut set.
{{e, f}, {f, g}} is another cut set.
{{b, c}, {c, d}} is another cut set.

NOTE: A set consisting of only a cut edge or bridge is always a cut set.

Theorem: A vertex v in a connected graph G is a cutvertex if and only if there exists two vertices x and y in G such that every path between x and y passes through v.

4.8.2 Connectivity

Edge connectivity: Let G be connected graph. Then the edge connectivity of G is the minimum number of edges whose removal disconnects the graph.

The edge connectivity is denoted by $\lambda(G)$

1. If G is disconnected graph, $\lambda(G) = 0$.
2. If G is a connected graph with a bridge, then $\lambda(G) = 1$.
3. $k(K_n) = n - 1$, where K_n is the complete graph of n vertices.
4. The vertex connectivity of a graph is one if and only if it has a cut vertex.

5. $k(C_n) = 2$, where C_n is the cycle graph of n vertices.
6. A connected graph is said to be separable if its vertex connectivity is one. All other connected graphs are called non separable.
7. In a separable graph, the vertex whose removal disconnects the graph is a cut vertex.
8. In a tree, every vertex with degree greater than one is a cut-vertex.

Theorem: The edge connectivity of a graph cannot exceed the degree of the vertex with smallest degree in G.

Theorem: The vertex connectivity of a graph G is always less than or equal to its edge connectivity. i.e. $k(G) \leq \lambda(G)$.

Definition: A graph is said to be ***n*-connected** if $k(G) = n$ and n-line connected if $\lambda(G) = n$. A 1-connected graph is the same as separable graph.

Theorem: $k(G) \leq \lambda(G) \leq 2e/n$ and max. vertex connectivity possible = $\lfloor 2e/n \rfloor$.

Example: for a graph with 8 vertices and 18 edges, we can achieve a vertex connectivity as high as four $\left(= \left\lfloor \frac{2\times 18}{8} \right\rfloor = 4\right)$.

Example - 4.6 What is the number of edges in the graph Q_7?

Solution:

We know that, Q_n is hasse diagram of boolean algebra. Q_n is always a n-regular graph. There are 2^n vertices in Q_n.

Hence, $$n \times 2^n = 2e$$

i.e. $$e = n \times 2^{n-1}$$

Hence is Q_7 we have $7 \times 2^6 \rightarrow 448$ edges.

Example - 4.7 15 vertices and 4 components in the graph. What is the minimum number of edges?

Solution:

We know that, $$n-k \leq e \leq \frac{(n-k+1)(n-k)}{2}$$

$$n-k \leq e$$

$\therefore$ $$15-4 \leq e$$

$$11 \leq e$$

Therefore minimum number of edges for the graph with 15 vertices and 4 components are 11.

Example - 4.8 What is the maximum number of edges with 17 vertices and 6 components?

Solution:

$$e \leq \frac{(n-k+1)(n-k)}{2}$$

$$e \leq \frac{(17-6+1)(17-6)}{2} \leq 6 \times 11 \leq 66$$

$\therefore$ Maximum edges are 66.

Example - 4.9 How many cut vertices in tree with '20' nodes and '9' leaf nodes?

Solution:

In tree all the internal nodes are cut vertices

$$n = i + l$$

$$20 = i + 9 \Rightarrow i = 11$$

$\therefore$ 11 cut vertices are tree.

Example - 4.10 In a computer lab, 50 computer and 75 cables are there. How many ways can we connect to have strongest connectivity?

Solution:

We know that complete graph (k_n) gives the strongest connectivity.

But we do not have $\frac{50(50-1)}{2}$ cables.

Hence we can make as strong as possible from given cables.

$$\text{Vertex connectivity of } (K_G) \le \left\lfloor \frac{2e}{n} \right\rfloor \le \left\lfloor \frac{2\times 75}{50} \right\rfloor \le 3$$

Hence, given 50 computer and 75 cables, there are 3 computers which if removed will disconnect the network.

Example - 4.11 The order of the graph is 20 and it's size in 35. Then what is the maximum vertex connectivity possible?

Solution:

$$\left\lfloor \frac{2e}{n} \right\rfloor = \left\lfloor \frac{2\times 35}{20} \right\rfloor = 3 \text{ is the strongest connectivity.}$$

Example - 4.12 The number of vertices and edges in a graph are 10 and 15 respectively. What is the maximum vertex connectivity and edge connectivity, given that it's minimum degree is 2?

Solution:

We know two relations,

$$\text{Vertex connectivity} \le \left\lfloor \frac{2e}{n} \right\rfloor$$

This given vertex connectivity $\le \left\lfloor \frac{2\times 15}{10} \right\rfloor \le 3$. But we have more strong answer

Vertex connectivity (K_G) $\le$ Edge connectivity (λ_G) $\le \delta$ (minimum degree) $\le \left\lfloor \frac{2e}{n} \right\rfloor$

$\delta = 2$ is given which gives a stronger upper bound for vertex connectivity and edge connectivity.

$\therefore$

$$K_G \le \lambda_G \le 2 \le 3$$

So in this problem the maximum possible vertex and edge connectivity is 2.

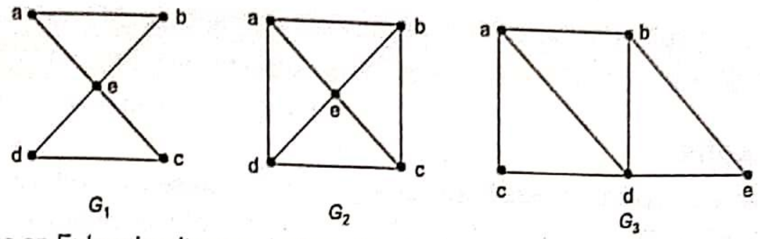
If minimum degree (δ) is not given then we take the average degree $\left(\left\lfloor \frac{2e}{n} \right\rfloor\right)$ as the upper bound for K_G as well as λ_G.

4.9 Euler Graphs

Euler Path: An open path in a graph G is called **Euler path** if it includes every edge exactly one. It is also called Euler Trail.

Euler Circuit: A circuit in a graph G that includes every edge exactly once is called an Euler circuit or Euler Cycle. A graph with an Euler cycle is called an **Euler Graph.**

Example:



Graph G_1 has an Euler circuit a e c d e b a. Graph G_2 does not have an Euler path or a circuit. Graph G_3 has an Euler path (a c d e b d a b) but not an Euler circuit.

Theorem: A connected multigraph has an Euler circuit if and only if each of vertices has even degrees. All vertices in G_1 are even degrees, hence it is an Euler graph.

Theorem: A connected multigraph has an Euler path but not an Euler circuit, if and only if it has exactly two vertices of odd degree. Such a path will begin at one of these odd vertices and end at the other.

G_3 has exactly two vertices of odd degree. These are vertices a and b respectively. Hence there is an Euler path but not circuit.

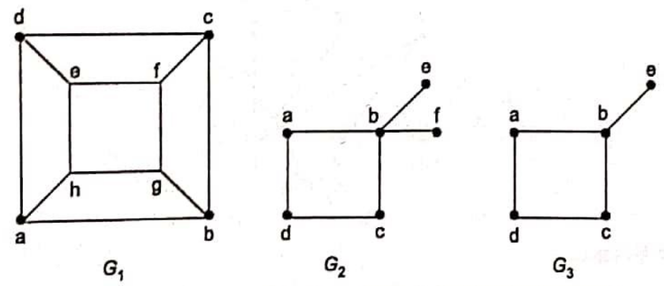
Theorem: A directed multigraph has an Euler circuit if and only if it is unilaterally connected and the indegree of every vertex in G is equal to its outdegree.

4.10 Hamiltonian Graphs

A **hamiltonian path** is a simple open path that visits all vertices of G, exactly once.

A hamiltonian circuit is a circuit that visits all vertices of G, exactly once.

A graph G is called a Hamiltonian Graph, If it contains a hamiltonian circuit.



Graph G_1 is a hamiltonian graph, since it contains a hamiltonian circuit, d c b a h g f e d.

A graph which contains a hamiltonian circuit will surely contain a hamiltonian path (by leaving out the last vertex of hamiltonian cycle). For example, G_1 contains the hamiltonian path d c b a h g f e. G_2 has neither a hamiltonian circuit, not a hamiltonian path. G_3 has a hamiltonian path, but not a hamiltonian circuit.

Hence G_3 is not a hamiltonian graph.

NOTE: K_n has a hamiltonian circuit whenever $n \geq 3$.

i.e. K_n, $n \geq 3$ is always a hamiltonian graph.

Theorem: In K_n, there are $(n-1)/2$ edge-disjoint hamiltonian circuits, if n is an odd number ≥ 3.

Theorem: A graph with n vertices and with no loops or parallel edges (i.e. a simple graph) which has at least $1/2(n-1)(n-2) + 2$ edges is hamiltonian. (The converse may or may not be true).

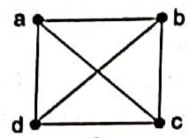
Dirac's Theorem: If G is a simple graph with n vertices, $n \geq 3$, such that the degree of every vertex in G is atleast $n/2$, then G has a hamiltonian circuit.

Ore's Theorem: If G is a simple graph with n vertices, $n \geq 3$, such that deg (u) + deg $(v) \geq n$ for every pair of nonadjacent vertices u and v in G, then G has a hamiltonian circuit.

NOTE: Unlike for Euler graphs, there as yet no necessary and sufficient condition developed for a graph to be a hamiltonian graph.

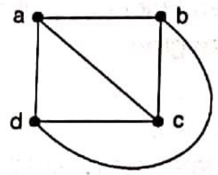
4.11 Planar Graphs

A graph G is said to be **planar** if there exists some geometric representation of G which can be drawn on a plane such that no two of its edges cross each other. The points of crossings are called crossovers. A representation of a graph drawn in a plane without edges crossing is called its planer representation or plane embedding.



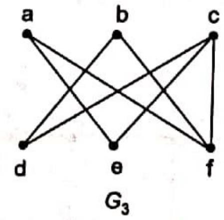
G_1

G_1 is a planer graph, since it has a planer representation, given below as G_2.



G_2

However the following graph G_3 is non-planar. i.e. it has no possible planar representation or embedding.

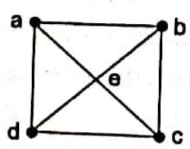


G_3

Euler's Formula for Planar Graphs

Theorem: If a connected planar graph G has n vertices, e edges and r regions, then $r = e - n + 2$.

Example:



Here, $n = 5$, $e = 8$ and $r = 5$

Since, $e - n + 2 = 8 - 5 + 2 = 5 = r$

$\therefore$ Euler's formula for planar graphs is verified.

Corollary: If a planar graph has k components, then $r = e - n + (k + 1)$

Corollary: If G is a connected, planar, simple graph with n (≥ 3) vertices and e edges, then $e \leq 3n - 6$.

Note: This corollary can be used to prove that a given graph is non-planar.

i.e. if in a simple and connected graph $e \not\leq 3n - 6 \Rightarrow$ graph is not planar.

Corollary: If G is a connected, planar, simple graph with n (≥ 3) vertices and e edges and no circuit of length 3, then $e \leq 2n - 4$. This corollary also can be used to prove that some graphs are non-planar.

Example-4.13 **A connected planar graph having 15 vertices and 12 regions. Find the number of edges connecting the vertices in this graph.**

(a) 15 (b) 20
(c) 25 (d) 30

Solution: (c)

Euler's formula: $V = E - R + 2$

$\Rightarrow$ $E = V + R - 2$

Given, V = 15 and R = 12

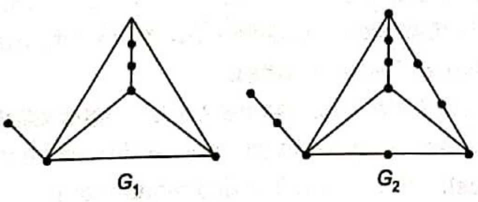
Number of edges (E) = 15 + 12 − 2
= 25

Kuratowski's Theorem (Necessary and sufficient condition for planarity) A graph is planar if and only if it contains no subgraph homeomorphic to K_5 or $K_{3,3}$.

NOTE: K_5 and $K_{3,3}$ are known as Kuratowski's two graphs. These two graphs are special since, K_5 is the non-planar graph with minimum number of vertices and $K_{3,3}$ is the non-planar graph with minimum number of edges.

Definition: Two graphs are said to be homeomorphic if both can be obtained from the same graph by elementary sub-divisions (insertion of new vertices of degree 2 into its edges or by merger of edges in series).

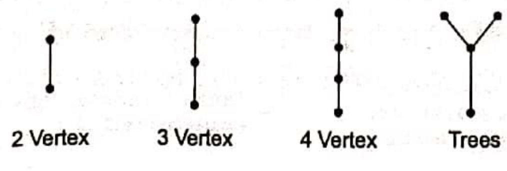
Example: G_1 and G_2 below are homeomorphic.



4.12 Trees

A connected, acyclic graph is a tree. i.e. A connected graph with no circuits is a tree. Its edges are called branches.

A tree with only one vertex is a trivial tree, otherwise it is a non trival tree.

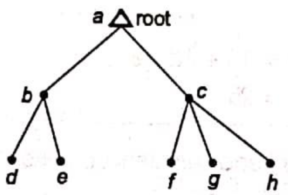


Properties of Trees:

1. A graph is a tree if and only if there is one and only one path between every pair of vertices.
2. A tree with n vertices has exactly $n-1$ edges.
3. If a connected graph G, has n vertices and $n-1$ edges, it is a tree.
   ∴ A graph with n vertices is called a tree if
   (a) It is connected and acyclic or
   (b) It is connected and has $n-1$ edges or
   (c) If it is acyclic and has $n-1$ edges or
   (d) If there is exactly one path between every pair of vertices in G.
4. Every edge of a tree is a cut set
5. Adding one edge to a tree forms exactly one cycle.
6. A tree is a minimally connected graph.

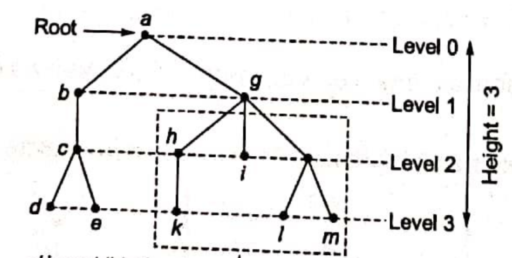
4.12.1 Rooted Trees

A tree in which a particulars vertex (called root) is distinguished from others is called a rooted tree.



A Rooted Tree

1. The **level** of a vertex is the number of edges along the unique path between it and the root. The level of the root is defined as 0. The vertices immediately under the root are said to be in level 1 and so on.
2. The **height** of a rooted tree is the maximum level to any vertex of the tree. * The **depth** of a vertex v in a tree is the length of the path from the root to v.
3. Given any internal vertex v of a rooted tree, the **children** of v are all those vertices that are adjacent to v and are one level further away from the root than v. If w is a child v, the v is called the **parent** of w, and two vertices that are both children of the same parent are called **siblings**.
4. If the vertex u has no children, then u is called a **leaf** (or a **terminal vertex**). If u has either one or two children, then u is called **an internal vertex**.
5. The **descendants** of the vertex u is the set consisting of all the children of u together with the descents of those children. 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.
   These terms are illustrated in given figure.

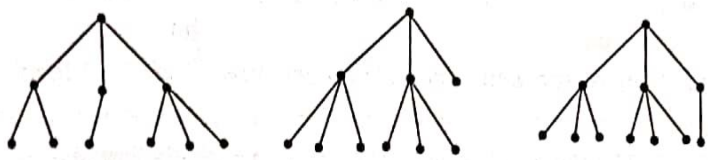


d is a child of *c*
c is a parent of *d*
d and *e* are sib lings

Vertices in enclosed region+ are descendants of *g*

Definition: A rooted tree is an ***m*-ary tree** if every internal vertex has at most m children. A m-ary tree is a **full *m*-ary tree** if every internal vertex has exactly m children. In particular, the 2-ary tree is called **binary tree.** A **full binary tree** is a binary tree in which each internal vertex has exactly two children.

A **complete *m*-ary tree** is an *m*-ary tree in which all its levels, except possibly the last have maximum number of nodes and if all nodes of the last level appear as far left as possible.



A 3-ary tree　　A full 3-ary tree　　A complete 3-ary tree

The relationship between i, the number of internal vertices and l, the number of leaves of a full *m*-ary can be proved by using the following theorem.

Theorem: A full *m*-ary tree with i internal vertex has $n = mi + 1$ vertices.

Proof: Since the tree is full *m*-ary, each internal vertex has *m* children and the number of internal vertex is i, the total number of vertex except the root is mi.

Therefore, the tree has $n = mi + 1$ vertices.

Since l is the number of leaves, we have $n = l + i$, Using the two equalities $n = mi + 1$ and $n = l + i$, the following results can easily be deduced.

A full *m*-ary tree with

(*i*) n vertices has $i = (n-1)/m$ internal vertices and $l = [(m-1)\,n + 1]/m$ leaves.

(*ii*) i internal vertices has $n = mi + 1$ vertices and $l = (m-1)\,i + 1$ leaves.

(*iii*) l leaves has $n = (ml-1)/\,(m-1)$ vertices and $i = (l-1)/(m-1)$ internal vertices.

Theorem:

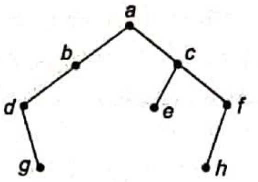
1. The maximum number of leaves in an *m*-ary tree of height h is m^h i.e., $l \le m^h$.
2. The maximum number of nodes in an *m*-ary tree of height h is $\dfrac{m^{h+1}-1}{m-1}$ i.e., $n \le \dfrac{m^{h+1}-1}{m-1}$.

4.12.2 Binary Tree

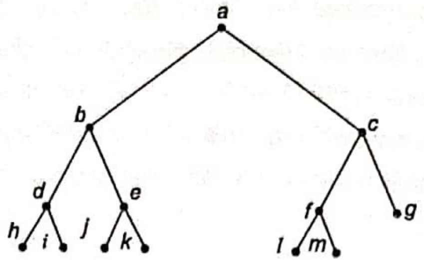
A binary tree is a rooted tree in which each vertex has atmost two children. Each child in a binary tree is designated either a **left child** or a **right child** (not both), and an internal vertex has at most one left and one right child. A **full binary tree** is a tree in which each internal vertex has exactly two children.

Given an internal vertex v of a binary tree, T, the **left subtree** of v is the binary tree whose root is the left child of v, whose vertex consist of the left child of v and all its descendants, and whose edges consist of all those edges of T that connect the vertices of the left subtree together. The **right subtree** of v is defined analogously.

Figure (a) below, is a binary tree and figure (b) is a full binary tree, since each of its internal vertices has exactly two children.



(a)



(b)

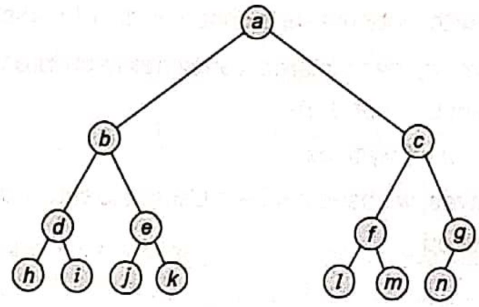
Theorem: If T is full binary tree with i internal vertices, then T has $i+1$ terminal vertices (leaves) and $2i+1$ total vertices.

Theorem: The maximum number of vertices on level n of a binary tree is 2^n where $n \geq 0$.

Theorem: The maximum number of vertices in a binary tree of depth d is $2^{d+1} - 1$ where $d \geq 1$.

Complete Binary Tree

If all the leaves of a full binary tree are at level d, then we call such a tree as a complete binary tree of depth d. A complete binary tree of depth of 3 is shown below.

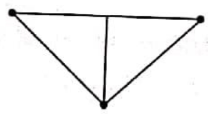


A complete binary tree

4.12.3 Spanning Trees

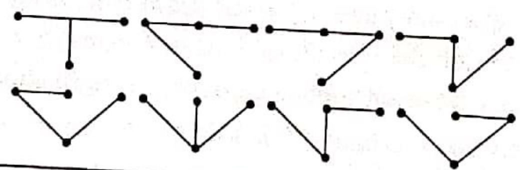
A subgraph T of G is called a spanning tree if T is a tree and if T includes every vertex of G i.e. V(T) = V(G).

Example-4.14 **Find all spanning trees of the graph G shown below.**



Solution:

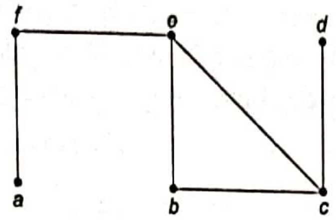
The graph G has four vertices and hence each spanning tree must have $4 - 1 = 3$ edges. Thus each tree can be obtained by deleting two of the five edges of G. This can be done in 10 ways, except that two of the ways lead to disconnected graphs. Thus there are eight spanning trees as shown in figure below.



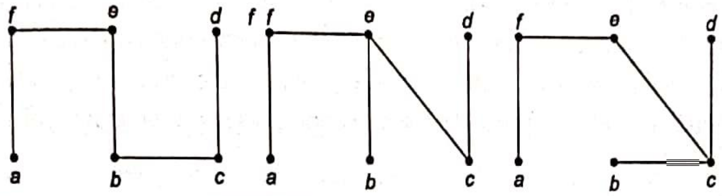
Theorem: A simple graph G has a spanning tree if and only if G is connected.

Example-4.15 Find all spanning trees for the graph G shown in figure. By removing the edges in simple circuits.

Solution:



The graph G has one cycle cbec and removal of any edge of the cycle gives a tree. There are three edges in the cycle and hence there are 3 spanning trees possible as shown below.



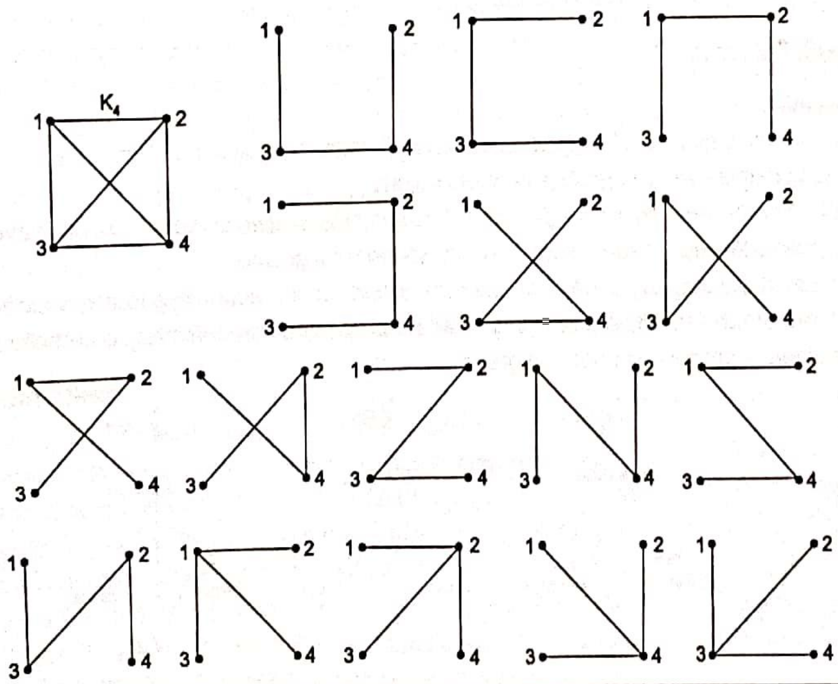
The number of different spanning trees on the complete graph K_n can be found from Cayley's theorem which is given below without any proof.

Cayley's Theorem: The complete graph K_n has n^{n-2} different spanning trees.

Example-4.16 Give all the spanning trees of K_4.

Solution:

Here $n = 4$, so there will be $4^{4-2} = 16$ different spanning trees. All the spanning trees of K_4 are shown in figure.



Weighted Graph

A **weighted graph** is a graph G in which each edge *e* has been assigned a non-negative number $w(e)$, called the weight (or length) of *e*.

Figure below shows a weighted graph. The weight (or length) of a path in such a weighted graph G is defined to be the sum of the weights of the edges in the path. Many optimisation problems amount to finding, in a suitable weighted graph, a certain type of subgraph with minimum (or maximum) weight.

4.12.4 Minimal Spanning Trees

Let G be a weighted graph. A minimal spanning tree of G is a spanning tree of G with minimum weight.

Algorithm for Minimal Spanning Trees

There are several methods available for actually finding a minimal spanning tree in a given graph.

Two algorithms due to Kruskal and Prim for finding a minimal spanning tree of a connected weighted graph where no weight is negative are available. These algorithms are example of greedy algorithms. A greedy algorithm is a procedure that makes an optimal choice at each of its steps without regard to previous choices.

4.13 Enumeration of Graphs

1. The number of simple, labeled graphs with n vertices and e edges is $\begin{pmatrix} \frac{n(n-1)}{2} \\ e \end{pmatrix} = {}^{\frac{n(n-1)}{2}}C_e$.
2. The number of simple, labeled graphs of n vertices is $= 2^{n(n-1)/2}$.
3. The number of labeled trees with n vertices $(n \geq 2) = n^{n-2}$.
4. The number of different rooted, labeled trees with n vertices is n^{n-1}
5. Enumerating unlabeled trees is more complicated due to isomorphism and involves the use of generating functions and partitions.

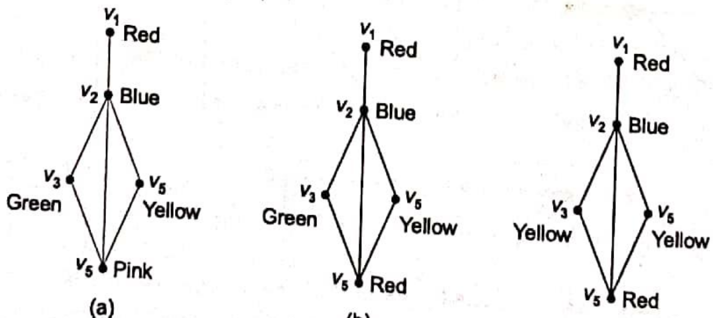
4.13.1 Graph Coloring

Chromatic Number

The chromatic number of a graph G written K(G) is the minimum number of colors needed to label the vertices so that adjacent vertices receive different colors.

Painting all the vertices of a graph with colors such that no two adjacent vertices have the same color is called the **proper coloring** (or sometimes simply coloring) of a graph.

A graph in which every vertex has been assigned a color according to a proper coloring is called a properly colored graph. Usually a given graph can be properly colored in many different ways. Figure above shows three different proper colorings of a graph.



(a) (b) (c)

The proper coloring which is of interest to us is one that requires the minimum number of colors. A graph G that requires k different colors for its proper coloring, and no less, is called a ***k*-chromatic graph**, and the number k is called the **chromatic number** of G. You can verify that the graph in above figure is 3-chromatic.

$K(G) = k$ means

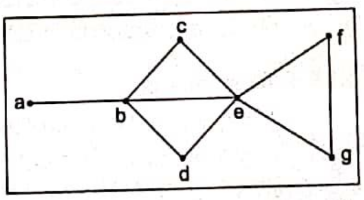
1. The graph can be colored with k colors.
2. The graph cannot be colored with fewer than k colors.

In coloring graphs there is no point in considering disconnected graphs. How we color vertices in one component of a disconnected graph has no effect on the coloring of the other components. Therefore, it is usual to investigate coloring of connected graphs only. All parallel edges between two vertices can be replaced by a single edge without affecting adjacency of vertices. Self-loops must be disregarded. Thus for coloring problems we need to consider only simple, connected graphs.

Some observations that follow directly from the definitions just introduced are:

1. A graph consisting of only isolated vertices is 1-chromatic.
2. A graph with one or more edges (not a self-loop, of course) is at least 2-chromatic (also called bichromatic).
3. A complete graph of n vertices is ***n*-chromatic**, as all its vertices are adjacent. Hence a graph containing a complete graph of r vertices is at least r-chromatic. For instance, every graph having a triangle is at least 3-chromatic.
4. 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. (This can be seen by numbering vertices 1, 2, ..., n in sequence and assigning one color to odd vertices and another to even, no adjacent vertices will have the same color. If n is odd, the nth and first vertex will be adjacent and will have the same color, thus requiring a third color for proper coloring.)
   This means $K(C_n) = 2$ if n is even and $K(C_n) = 3$ if n is odd
   Where, C_n is the cycle graph with n vertices.
5. **Theorem:** Every tree with 2 or more vertices is 2-chromatic.
6. **Theorem:** A graph with atleast one edge is 2-chromatic if and only if it has no circuits of odd length. i.e. bipartite graphs (which have no circuits of odd length) are 2-chromatic. Also, $K(K_{m,n}) = 2$.
7. **Theorem:** If d_{max} is the maximum degree of the vertices in a graph G, Chromatic number of $G \leq 1 + d_{max}$.
8. Apple and Haken proved the **4 color theorem** which says that every planar graph can be colored by at most 4 colors. That is chromatic number of every planar graph is 4 or less.
   Proper coloring of a given graph is simple enough, but a proper coloring with the minimum number of colors is, in general, a difficult task. In fact, there has not yet been found a simple way of characterizing a k-chromatic graph. (The brute-force method of using all possible combinations can, of course, always be applied, as in any combinatorial problem. But brute force is highly unsatisfactory, because it gets out of hand as soon as the size of the graph increases beyond a few vertices).

4.13.2 Independent Sets



A proper coloring of a graph naturally induces a partitioning of the vertices into different subsets. For example, the coloring in figure (graph coloring: previous page) produces the partitioning $\{v_1, v_4\}$, $\{v_2\}$ and $\{v_3, v_4\}$.

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

A set of vertices in a graph is said to be an **independent set** of vertices or simply an independent set (or an internally stable set) if no two vertices in the set are adjacent. For example, in figure below, {a, c, d} is an independent set. A single vertex in any graph constitutes an 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 figure above, is a maximal independent set. The set {b, f} is another maximal independent set. The set {b, g} is a third one. From the preceding example, it is clear that a graph, in general, has many maximal independent sets, and they may be of different sizes. Among all maximal independent sets, one with the largest number of vertices is often of particular interest.

The number of vertices in the largest independent set of a graph G is called, the **independence number** (or coefficient of internal stability), β(G).

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

$$\beta(G) \geq \frac{n}{k}$$

NOTE: A graph G is *k*-partite if V(G) can be expressed as union of *k* independent sets.

4.13.3 Dominating Sets

A **dominating set** (or an externally stable set) in a graph G is a set of vertices that dominates every vertex *v* in G in the following sense: Either *v* is included in the dominating set or is adjacent to one or more vertices included in the dominating set.

For instance, the vertex set {b, g} is a dominating set in Figure above. So is the set {a, b, c, d, f } a dominating set.

A dominating set need not be independent. For example, the set of all its vertices is trivially a dominating set in every graph.

In many applications one is interested in finding minimal dominating sets defined as follows:

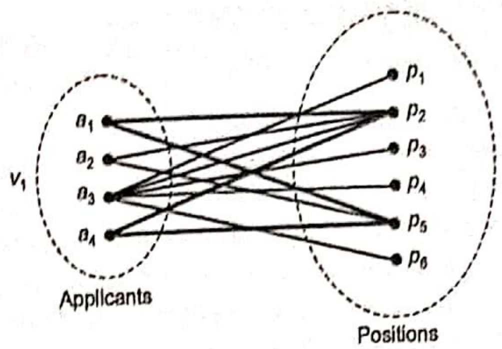
A **minimal dominating set** is a dominating set from which no vertex can be removed without destroying its dominance property. For example, in above figure {b, e,} is a minimal dominating set. And so is {a, c, d, f}.

Observations that follows from these definitions are:

1. Any one vertex in a complete graph constitutes a minimal dominating set.
2. Every dominating set contains at least one minimal dominating set.
3. A graph may have many minimal dominating sets, of different sizes. [The number of vertices in the smallest minimal dominating set of a graph G is called domination number, α (G).
4. A minimal dominating set may or may not be independent.
5. Every maximal independent set is a dominating set. For if an independent set does not dominate the graph, there is at least one vertex that is neither in the set nor adjacent to any vertex in the set. Such a vertex can be added to the independent set without set without destroying its independence. But then the independent set could not have been maximal.
6. An independent set has the dominance property only if it is a maximal independent set. Thus an independent dominating set is the same as a maximal independent set.
7. In any graph G, a(G) ≤ β (G).

4.13.4 Matchings

Suppose that four applicants a_1, a_2, a_3 and a_4 are available to fill six vacant positions P_1, P_2, P_3, P_4, P_5, and P_6. Applicant a_1 is qualified to fill position P_2 or P_5. Applicant a_2 is qualified to fill position P_2 or P_5. Applicant a_3 is qualified for P_1, P_2, P_3, P_4 or P_6. Applicant a_4 can fill jobs P_2 or P_5. This situation is represented by the graph in figure below. The vacant positions and applicants are represented by vertices. The edges represent the qualifications of each applicant for filling different positions. The graph clearly is bipartite, the vertices falling into two sets $V_1 = \{a_1, a_2, a_3, a_4\}$ and $V_2 = \{p_1, p_2, p_3, p_4, p_5, p_6\}$.



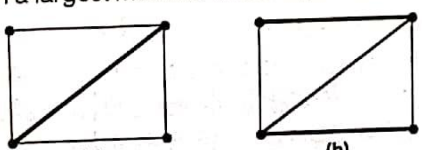
Bipartite graph

The questions one is most likely to ask in this situation are : is it possible to hire all applicants and assign each a position for which he is suitable. If the answer is no, what is the maximum number of positions that can be filled from the given set of applicants?

This is a problem of matching (or assignment) of one set of vertices into another. More formally, a **matching** in a graph is a subset of edges in which no two edges are adjacent. A single edge in a graph is obviously a matching.

A **maximal matching** is a matching to which no edge in the graph can be added. For example, in a complete graph of three vertices (i.e., a triangle) any single edge is a maximal matching.

The edges shown by heavy lines in above figure are two maximal matchings. Clearly, a graph may have many different maximal matchings, and of different sizes. Among these, the maximal matching with the largest number of edges are called the **largest maximal matching**. In figure (b), a largest maximal matching is shown in heavy lines. The number of edges in a largest maximal matching is called the **matching number** of the graph.



(a) (b)

Although matching is defined for any graph, it is mostly studied in the context of bipartite graphs, as suggested by the introduction to this section.

In a bipartite graph having a vertex partition V_1 and V_2, a **complete matching** of vertices in set V_1 into those in V_2 is a matching in which there is one edge incident with every vertex in V_1. In other words, every vertex in V_1 is matched against some vertex in V_2. Clearly, a complete matching (if it exists) is a largest maximal matching, whereas the converse is not necessarily true.

For the existence of a complete matching of set V_1 into set V_2, first we must have at least as many vertices in V_2 as there are in V_1. In other words there must be at least as many vacant positions as the number of applicants if are to be hired. This condition, however, is not sufficient. For example, the above figure, although there are six positions and four applicants, a complete matching does not exist. Of the three applicants a_1, a_2 and a_4 each qualifies for the same two positions P_2 and P_5 and therefore one of the three applicants cannot be matched.

This leads us to another necessary condition for a complete matching: Every subset of r verties in V_1 must collectively be adjacent to at least r vertices in V_2, for all values or $r = 1, 2 \ldots, |V_1|$. This condition is not satisfied in above figure. The subset $\{a_1, a_2, a_4\}$ of three vertices has only two vertices P_2 and P_5 adjacent to them. That this condition is also sufficient for existence of a complete matching is indeed surprising. Theorem below is a formal statement and proof of this result.

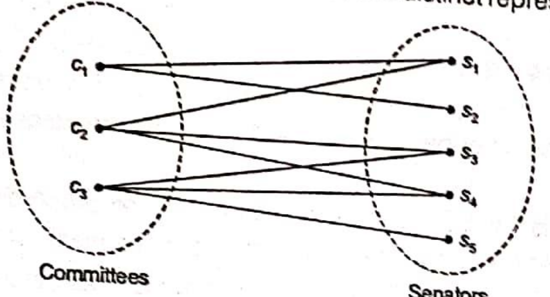
NOTE: *Theorem:* A complete matching of V_1 into V_2 in a bipartite graph exists if and only if every subset of r vertices in V_1 is collectively adjacent to r or more vertices in V_2 for all values of r.

Let us illustrate this important theorem with an example:

Problem of Distinct Representatives: Five senators s_1, s_2, s_3, s_4 and s_5 are members of three committees c_1, c_2 and c_3. The membership is shown in figure below. One member from each committee is to be represented in a super committee. Is it possible to send one distinct representative from each of the committees?

This problem is one of finding a complete matching of a set V_1 into set V_2 in a bipartite graph. Let us use theorem above and check if r vertices from V_1 are collectively adjacent to at least r vertices from V_2, for all values of r. The result is shown in the Table below (ignore the last column for the time being).

Thus for this example the condition for the existence of a complete matching is satisfied as stated in Theorem. Hence it is possible to form the supercommittee with one distinct representative from each committee.



Membership of committees

	V_1	V_2	$r - q$
$r = 1$	$\{c_1\}$	$\{s_1, s_2\}$	-1
	$\{c_2\}$	$\{s_1, s_3, s_4\}$	-2
	$\{c_3\}$	$\{s_3, s_4, s_5\}$	-2
$r = 2$	$\{c_1, c_2\}$	$\{s_1, s_2, s_3, s_4\}$	-2
	$\{c_2, c_3\}$	$\{s_1, s_3, s_4, s_5\}$	-2
	$\{c_3, c_1\}$	$\{s_1, s_2, s_3, s_4, s_5\}$	-3
$r = 2$	$\{c_1, c_2, c_3\}$	$\{s_1, s_2, s_3, s_4, s_5\}$	-2

In above table, $r = |V_1|$ and $q = |V_2|$

The problem of distinct representatives just solved was a small one. A larger problem would have become unwieldy. If there are M vertices in V_1. Theorem requires that we take all $2^M - 1$ non-empty subsets of V_1 and find the number of vertices of V_2 adjacent collectively to each of these. In most cases, however, the following simplified version of the theorem will suffice for detection of a complete matching in any large graph.

NOTE: *Theorem:* In a bipartite graph a complete matching of V_1 into V_2 exists if (but not only if) there is a positive interger m for which the following condition is satisfied:
Degree of every vetex in V_1 ³ m³ degree of every vertex in V_2.

In the bipartite graph of above figure. Degree of every vertex in $V_1 \geq 2 \geq$ degree of every vertex in V_2. Therefore, there exists a complete matching.

In the bipartite graph of figure in previous page (applicants and positions), no such number is found, because the degree of $p_2 = 4 >$ degree of a_1.

It must be emphasized that the condition of Theorem above, is a sufficient condition and not necessary for the existence of a complete matching. It will be instructive for the reader to sketch a bipartite graph that does not satisfy the theorem and yet has a complete matching.

If one fails to find a complete matching, he is most likely to be interested in finding a maximal matching, that is, to pair off as many vertices of V_1 with those in V_2 as possible. For this purpose, let us define a new term called deficiency, δ (G), of a bipartite graph G.

A set of r vertices in V_1 is collectively incident on, say, q vertices of V_2. Then the maximum value of the number $r-q$ taken over all values of $r = 1, 2, \ldots$ and all subsets of V_1 is called the deficiency $\delta(G)$ of the bipartite graph G.

Theorem above, expressed in terms of the deficiency, states that a complete matching in a bipartite graph G exists if and only if $\delta(G) \leq 0$. For example, the deficiency of the bipartite graph in above figure is -1 (the largest number in the last column of above table). It is suggested that you prepare a table for the graph in previous page (applicants and positions) similar to above table and verify that the deficiency is $+1$ for this graph.

The next theorem gives the size of the maximal matching for a bipartite graph with a positive deficiency.

Theorem: The maximal number of vertices in set V_1, that can be matched into V_2 is equal to number of vertices in $V_1 - \delta(G)$.

The size of a maximal matching in the problem of applicants and positions, using above theorem, is obtained as follows:

Number of vertices in $V_1 - \delta(G) = 4 - 1 = 3$

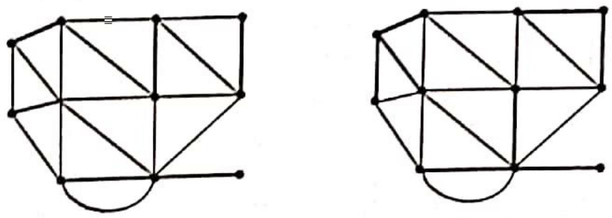
4.13.5 Coverings

In a graph G, a set g of edges is said to 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**, a **covering subgraph**, or simply a **covering** of G.

For example a graph G is trivially its own covering. A spanning tree in a connected graph (or a spanning forest in an unconnected graph) is another covering.

A Hamiltonian circuit (if it exists) in a graph is also a covering. Just any covering is too general to be of much interest. We have already dealt with some coverings with specific properties, such as spanning trees and Hamiltonian circuits. In this section we shall investigate the **minimal covering** - a covering from which no edge can be removed without destroying its ability to cover the graph.



(a) e = 6 (b) e = 8

A Graph and two of its minimal coverings

In above figure a graph and two of its minimal coverings are shown in heavy lines. The following observations should be made:

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

6. A graph, in general, has many minimal coverings, and they may be of different sizes (i.e., consisting of different numbers of edges). The number of edges in a minimal covering of the smallest size is called the covering number of the graph.

Theorem: A covering g of a graph is minimal if and only if g contains no paths of length three or more.

Suppose that the graph in above figure represented the street map of a part of a city. Each of the vertices is a potential trouble dot and must be kept under the surveillance of a patrol car. How will you assign a minimum number of patrol cars to keep every vertex covered? The answer is a smallest minimal covering. The covering (a) shown in above figure is an answer, and it requires six patrol cars. Clearly, since there are 11 vertices and no edge can cover more than two, less than six edges cannot cover the graph.

Summary

- A vertex with zero degree is called a **lone vertex** or isolated vertex and a vertex with exactly one degree is called a **pendent** vertex or end vertex.
- A complete graph is a simple graph with maximum number of possible edges.
- The number of edges in $K_n = \dfrac{n(n-1)}{2} = nC_2$

  Example: K_5 has 5 vertices and $\dfrac{5(5-1)}{2} = 10$ edges
- If a simple graph has nC_2 edges it is complete.
- Maximum edges possible in an n-vertex simple graph is $\dfrac{n(n-1)}{2}$.
- Isomorphic simple graphs must have:
  - Same no of vertices
  - Same no of edges
  - Degrees of corresponding vertices must be same
  - Number of simple circuits of a certain length must be same in both graphs. All the above are called invariants in an isomorphism.
- If G is a simple graphs of n vertices, then $G \cup \overline{G}$ is K_n, the complete graph on n vertices.
- When $k=1$, a simple graph with n vertices can have atmost $(n-1)n/2$ edges which is the maximum possible edges in a connected simple graph.
- Strongly connected $\Rightarrow$ Unilateral as well as weakly connected but converse is not true
  Unilaterally connected $\Rightarrow$ Weakly connected but converse is not true
  If a diagraph is not even weakly connected, then such a graph will be a disconnected graph.
- K_5 and $K_{3,3}$ are known as Kuratowski's two graphs. These two graphs are special since, K_5 is the non planar graph with minimum number of vertices and $K_{3,3}$ is the non planar graph with minimum number of edges.
- Every tree with 2 or more vertices is 2-chromatic.
- A graph with atleast one edge is 2-chromatic if and only if it has no circuits of odd length. i.e. bipartite graphs (which have no circuits of odd length) are 2-chromatic. Also, $K(K_{m,n}) = 2$.

- If d_{max} is the maximum degree of the vertices in a graph G, Chromatic number of $G \leq 1 + d_{max}$.
- A complete matching of V_1 into V_2 in a bipartite graph exists if and only if every subset of r vertices in V_1 is collectively adjacent to r or more vertices in V_2 for all values of r.
- In a bipartite graph a complete matching of V_1 into V_2 exists if (but not only if) there is a positive interger m for which the following condition is satisfied: Degree of every vetex in $V_1{}^3\, m^3$ degree of every vertex in V_2.

Student's Assignment

Q.1 The minimum number of edges in a connected cyclic graph on 2 vertices is

(a) 1 (b) 2
(c) 3 (d) None of these

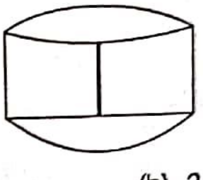
Q.2 If the following degree vertex (5, 3, 2, 4, 3), can't form a simple graph then which of the following degree vertex should be added to make it form a simple graph?

(a) 1 (b) 5
(c) 2 (d) Both (a) and (b)

Q.3 How many edges are possible in a forest with n vertices and k trees?

(a) $n - k + 1$ (b) $n - k - 1$
(c) $n - k$ (d) Insufficient data

Q.4 The minimum number of colours required to colour the following graph is

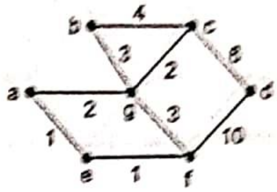


(a) 1 (b) 2
(c) 3 (d) 4

Q.5 What is the maximum length of cycles in a diagraph of partial order on A having n elements?

(a) 2 (b) 3
(c) 1 (d) n

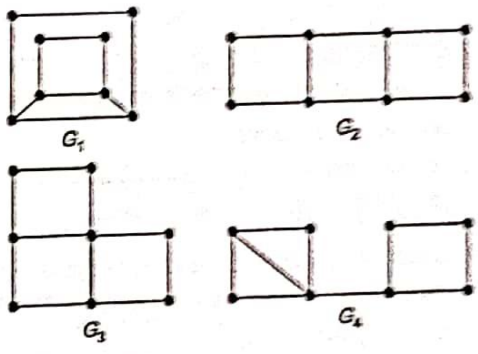
Q.6 Consider the following graph.



While constructing the minimum spanning tree of the above graph using Kruskal's algorithm, which of the following is a possible order in which the edges are added to the minimum spanning tree?

(a) (a,e), (a,g), (g,f), (b,c), (c,d), (d,f)
(b) (a,e), (e,f), (a,g), (g,c), (g,f), (b,g)
(c) (a,e), (e,f), (a,g), (g,c), (b,g), (b,c)
(d) (e,f), (a,e), (g,c), (a,g), (b,g), (c,d)

Q.7 Determine which pairs G_i, G_j of the graphs below are isomorphic



(a) $(G_1; G_2)$
(b) $(G_1; G_2)$ and $(G_1; G_3)$
(c) $(G_1; G_2)$ and $(G_2; G_3)$
(d) $(G_1; G_2)$ and $(G_3; G_4)$

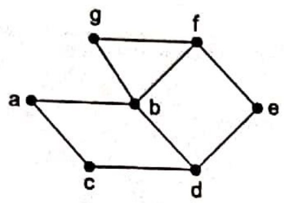
Q.8 Which of the following statements are true?

(i) Euler's problem can be solved in polynomial time (*ii*) Hamiltan's problem is believed to lie in class NP (Non-deterministic polynomial)

(*iii*) Traveling saleman problem can also be proved to be in NP

(a) Only (*i*) and (*iii*) are true
(b) Only (*i*) and (*ii*) are true
(c) Only (*ii*) and (*iii*) are true
(d) All (*i*), (*ii*) and (*iii*) are true

Q.9 Consider the following graph:



Which of the following statements is true about the graph?

(a) It has an Eulerian path, but not Eulerian cycle
(b) It has an Eulerian cycle, but not Eulerian path
(c) It has both Eulerian path and Eulerian cycle
(d) It doesn't have an eulerian path or Eulerian cycle

Q.10 Which of the following statements is/are false?

(*i*) An edge e of a graph G is cut edge if e is not part of any cycle in g.

(*ii*) A connected graph G is Euler if and only if the degree of every vertex is odd.

(*iii*) A relation on a set P is said to be partial if it is reflexive, symmetric and transitive.

(*iv*) All of the above

(a) (*ii*) only
(b) (*i*) and (*ii*)
(c) (*ii*) and (*iii*)
(d) (*iv*)

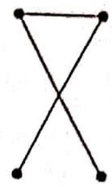
Q.11 Suppose G = G (V, E) has five vertices. Find the maximum number *m* of edges in E if G is a simple graph.

(a) 10
(b) 15
(c) 20
(d) None of these

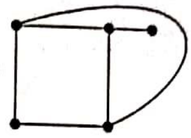
Q.12 Suppose G = G (V, E) has five vertices, find the maximum number *m* of edges in E if G is a multigraph

(a) Finite
(b) Infinite
(c) ${}^{m}C_{2}$
(d) None of the above

Common Data Questions (13 to 15):

(*i*)


(*ii*)


(*iii*)


(*iv*)


Q.13 From the above graphs, which of them are connected.

(a) (*i*) and (*iii*)
(b) (*ii*) and (*iii*)
(c) (*iii*) and (*iv*)
(d) (*iv*) and (*i*)

Q.14 Which of the graphs are multigraphs (i.e., have no loops, multiple edges allowed)

(a) (*i*), (*ii*), (*iii*)
(b) (*ii*), (*iii*), (*iv*)
(c) (*iii*), (*iv*), (*i*)
(d) (*iv*), (*i*), (*ii*)

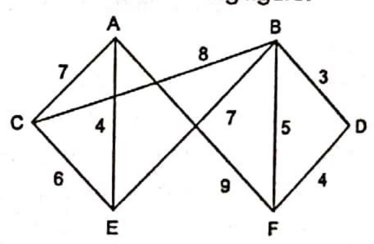
Q.15 Which of the above are simple graphs?

(a) (*i*), (*ii*) and (*iii*)
(b) (*ii*) and (*iii*)
(c) (*iii*) and (*iv*)
(d) None of these

Q.16 Number of edges of a complete binary tree with 16 leaf nodes is

(a) 14
(b) 30
(c) 32
(d) 28

Q.17 Find the length of minimal spanning tree for graph represented in the following figure.

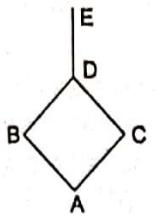


(a) 28
(b) 55
(c) 24
(d) None of these

Q.18 If the degree of every non-pendent vertex in a tree is 3, then the number of vertex of the tree is.

(a) odd
(b) even
(c) odd or even
(d) such a tree is not possible

Q.19 Consider the following figure which of the following is true?



(a) There exists a Euler path but not Euler circuit
(b) There exists a Euler circuit
(c) Euler path is not possible
(d) None of the above

Q.20 How many edges are there in a complete graph having 12 nodes?

(a) 12 (b) 144
(c) 66 (d) None of these

Answer Key:

1. (b)	**2.** (d)	**3.** (c)	**4.** (c)	**5.** (c)
6. (d)	**7.** (a)	**8.** (d)	**9.** (a)	**10.** (c)
11. (a)	**12.** (b)	**13.** (a)	**14.** (a)	**15.** (a)
16. (b)	**17.** (c)	**18.** (b)	**19.** (a)	**20.** (c)

Student's Assignments Explanations

2. (d)

By using Hevell-Hakimi theorem:

$$(\cancel{5}43321)$$
$$(\cancel{3}2210)$$
$$(1100)$$

Since number of 1 are even. Hence 1 is possible.
Similarly by using Hevell-Hakimi theorem:

$$(\cancel{5}54332)$$
$$(\cancel{4}3221)$$
$$(\cancel{2}110)$$
$$(000)$$

Since number of 1 are even. Hence 5 is possible.
So both 1 and 5 are possible.

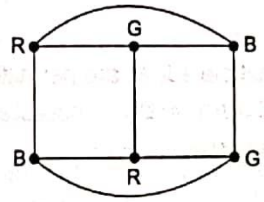
3. (c)

As the problem definition says 'n' vertices are divided in to 'k' trees. Let these trees are having $x_1, x_2, x_3, \ldots, x_k$, vertices respectively. The sum of $x_1, x_2, \ldots, x_k$ should be 'n' as we have only 'n' vertices in total. The definition of tree says that any tree says that any tree with 'n' vertices will have '$n - 1$' edges. So all the 'k' trees will have '$x_1 - 1 + x_2 - 1 + \ldots + x_k - 1$' edges

$$\text{Result} = x_1 - 1 + x_2 - 1 + \ldots + x_k - 1$$
$$= x_1 + x_2 + \ldots + x_k - k$$
$$= n - k$$

4. (c)

Let's say we have 3 colors namely R, G and B.

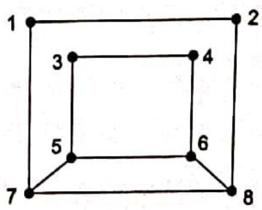


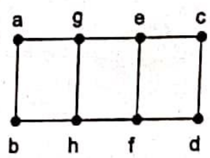
7. (a)

The degree sequence of the above graphs is:

G_1: (2, 2, 2, 2, 3, 3, 3, 3);
G_2: (2, 2, 2, 2, 3, 3, 3, 3);
G_3: (2, 2, 2, 2, 2, 3, 3, 4);
G_4: (2, 2, 2, 2, 2, 3, 3, 4).

G_1 is isomorphic to G_2 is shown by the following labeling:





1 corresponding to a, 2 to b, etc.

The differing degree sequences show neither G_3 nor G_4 can be isomorphic to either G_1 or G_2. Also G_3 is not isomorphic G_4 because G_4 has triangles (circuits of length 3) and G_3 does not.

9. (a)

The graph has no Euler cycle since f and d have odd degree.

However, since all vertices have even degrees except exactly 2 vertices namely f and d, therefore the graph does have an Eulerian path. One such eulerian path is f, b, g, f, e, d, b, a, c, d.

Other variants are possible, but in each variant the first and last vertex has to be f and d or d and f since these are the two vertices with odd degrees.

11. (a)

There are C(5, 2) = 10 ways of choosing two vertices from V to make an edge; hence $m = 10$.

12. (b)

Since multiple edges are permitted (in a multi-graph), G can have any number of edges between 2 vertices. Hence $m = \infty$.

13. (a)

Only (*i*) and (*iii*) are connected.

14. (a)

Only (*iv*) has a loop, (*i*), (*ii*) and (*iii*) are loop free.

15. (a)

Only (*i*) and (*ii*) are simple graphs. The multigraph (*iii*) has multiple edges and (*iv*) has multiple edges and a loop.

16. (b)

$$n = 2i + 1$$

Also $\quad n = \ell + i$

Since $\quad \ell = 16$

$$n = 16 + i = 2i + 1$$

$\Rightarrow \quad i = 15$

$$n = \ell + i = 16 + 15 = 31$$

$$e = n - 1 = 31 - 1 = 30$$

17. (c)

Use Kruskal's algorithm as given below:

- Arrange the edges in ascending order of weight.
- Add selectively the edges, such that addition of edge doesn't form a cycle.

The edges added are in the following order BD, DF, AE, CE, BE.

Total length is 24.

18. (b)

Let n be the number of vertices of the tree and let p be the number of pendent vertices.

Now, $\quad \Sigma deg = 2e$

In a tree $e = n - 1$

$\therefore \quad \Sigma deg = 2(n - 1)$

Since in this tree every non-pendent vertex has degree 3, and since pendent vertices have degree 1 always, we can say that

$$p + 3(n - p) = 2(n - 1)$$

Solving which we get

$$n = 2p - 2 = 2(p - 1)$$

which is even.

19. (a)

Since there are exactly 2 vertices (E and D) in this graph with odd degree, this graph has an euler path but not an euler circuit.

20. (c)

The number of edges in K_n is

$$^nC_2 = \frac{n(n-1)}{2}$$

K_{12} has $\frac{12(12-1)}{2} = 6.11 = 66$ edges