Module 4: Graph Theory	08 Hours
4.1 Types of graphs, Graph Representation, Sub graphs, Operations on Graphs, Walk, Path, Circuit, Connected Graphs, Disconnected Graph, Components.	
4.2 Homomorphism and Isomorphism of Graphs, Euler and Hamiltonian Graphs, Planar Graph, Cut Set, Cut Vertex, Applications.	

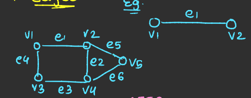
GRAPHS

I: BASICS

GRAPHS: definition

A graph G is mathematical structure consisting of two sets V and E where V is a non-empty set of vertices and E is non-empty set of edges

Eg.



BASIC TERMINOLOGIES

(1) **Trivial graph**

A graph consisting only one vertex and no edge

Eg. o v1

(2) **NULL graph**

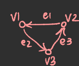
A graph consisting n vertices and NO EDGE is a null graph

Eg. o v1 o v2

(3) **Directed Graph**

A graph consisting direction of edges is called as directed graph

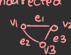
Eg.



(4) **Undirected Graph**

A graph which is not directed is undirected graph

Eg.



(5) **Self loop in Graph**

If edge having same vertex as both its end vertices is called as a self loop.

(जिथून सुरू तिथेच समाप्त)

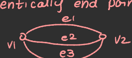
Eg

(6) **Proper edge**

An edge which is not self loop is called proper edge.

(7) **Multi edge.**

Collection of two or more edges having identically end point



e1, e2, e3 are multiedges with same endpoints v1 & v2

(8) **Simple Graph**

A graph does not contain any self loop and multiedge is Simple graph.

Eg.

(9) **Multigraph**

A graph not containing any self loop but containing a multiedge is called as multigraph (atleast one)

Eg.

(10) **Pseudo graph**

A graph containing both multiedge and self loop is called pseudo graph.

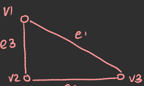
Eg.

(11) **Incidence and Adjacency.**

Let e be an edge joining two vertices v1 and v2, then e is said as Incidence of v1 & v2 and v1 and v2 are said to be Adjacency of e

Two vertices are said to be adjacent if there exists an edge joining those vertices.

Eg.

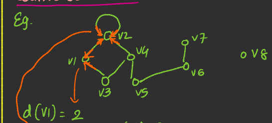


e1 is incidence of v1 & v3
e2 is incidence of v2 & v3
{e1 is not incidence of v1 & v2 or v2 & v3}
v1 & v2 is adjacency of e3

(12) **Degree of vertex.**

The degree of vertex V in a graph G {written as d(V)} is equal to the number of edges which are incident on V with self loop countered twice.

Eg.



d(v1) = 2
d(v2) = 4
d(v3) = 2
d(v4) = 3
d(v7) = 1
d(v8) = 0

(13) **Isolated Vertex**

A vertex having 0 degree is called as an isolated vertex

Eg. above
v8 is isolated vertex

Pendant Vertex

A vertex having 1 degree is called as Pendant vertex

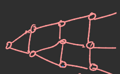
Eg.
v7 is pendant vertex.

(14) **Finite and Infinite Graphs.**

Graph with finite no. of vertices.
Graph with infinite no. of vertices.

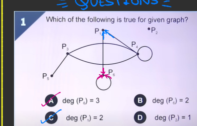
Eg.

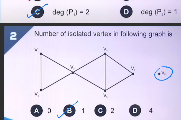
Finite

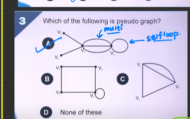


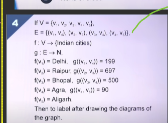
Infinite graph

QUESTIONS







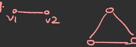


II: TYPES OF GRAPH

(1) **Complete Graph**

A simple connected graph is said to be complete if each vertex is connected to every other vertex (basically no open chain)

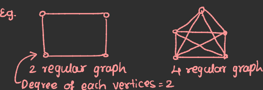
Eg.



(2) **Regular Graph.**

A graph is said to be regular if every vertex has the same degree. If degree of each vertex is k, then Graph G is called as k-regular graph.

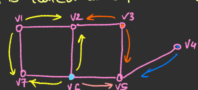
Eg.



(3) **Bigraph (or Bipartite)** ~~not in syllabus~~

If the vertex set V of graph G can be "partitioned" into two non-empty disjoint subsets X and Y, in such a way that edge of G has one end in X and other end in Y, Then G is called as Bipartite/Bigraph

Eg.



here $V = \{v_1, v_2, v_3, v_4, v_5, v_6, v_7\}$

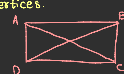
$X = \{v_1, v_6, v_3, v_4\}$

$Y = \{v_2, v_7, v_5\}$

CONNECTED GRAPH

An undirected graph is said to be connected if there is a path between every two vertices.

Eg.



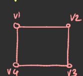
NOTE: If graph is connected, then it CANNOT be BIPARTITE!

(4) **Complete Bipartite Graph** ~~Not in syllabus~~

If every vertex in X is disjoint

If X and Y condition m & n vertices, then this graph is denoted as K_{mn}

Eg.



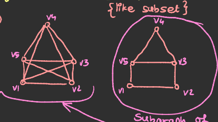
$X = \{v_1, v_3\}$
$Y = \{v_2, v_4\}$

each element in X is connected to every element in Y ⇒ Complete Bipartite

(5) **Subgraph**

Let G(V, E) be a graph, let V' be a subset of V and let E' be a subset of E whose end point belong to V'. Then G(V', E') is graph called as subgraph of G(V, E) {like subset}

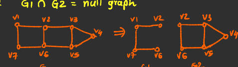
Eg.



DECOMPOSITION OF GRAPH

A graph is said to be decomposed into two subgraphs G_1 and G_2 if $G_1 \cup G_2 = G$ and $G_1 \cap G_2$ = null graph

Eg.



= Union

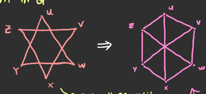
Intersection
=
common
=

[o v2, o v6] # NULL GRAPH

COMPLEMENT OF GRAPH

The complement of a graph G is defined as a simple graph with the same vertex set as G and where any two vertices (u & v) are adjacent only when they are not adjacent in G

Eg.

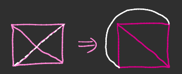


Jithe lines astil te kadhun taka & Jithe lines nastil tithe lines draw kara ⇒ # complement of graph

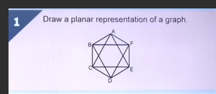
PLANARE GRAPH

A graph which can be drawn in the plane, so that none of its edge do not cross each other is called as planare graph

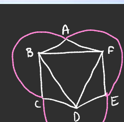
Eg.



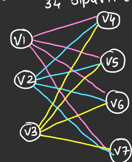
planare graph



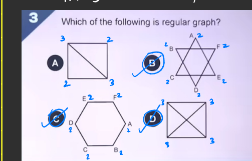
⇒

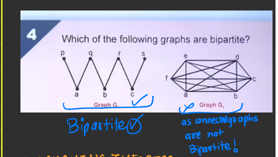


Q.2 Draw K_{34} bipartite graph



m : 3   n : 4





HANDSHAKING THEOREM

The sum of degree of vertices of graph G is equal to twice the total number of edges in G

Proof:

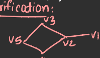
Consider graph G with 'e' edges & 'n' vertices, $v_1, v_2, v_3 \dots v_n$

Each edge contribute 2 degree in G

∴ sum of degree of all vertices

$$\sum_{i=1}^{n} \deg(v_i) = 2e$$

Verification:



$\deg(v_1) = 1$
$\deg(v_2) = 3$
$\deg(v_3) = 2$
$\deg(v_4) = 2$
$\deg(v_5) = 2$

$\sum_{i=1}^{4} \deg(v_i) = 10$

e: no. of edges = 5

∴ Hence Verified!

GRAPH REPRESENTATION

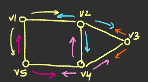
Matrix Representation:

① Adjacency Matrix

Let a_{ij} denote the number of edges (v_i, v_j) then $A = [a_{ij}]_{m \times m}$ is called adjacency matrix of G if

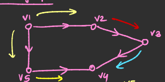
$$a_{ij} = \begin{cases} 1 & \text{if } (v_i, v_j) \text{ has an edge} \\ 0 & \text{otherwise} \end{cases}$$

Eg.



$$A = \begin{array}{c} \\ v1 \\ v2 \\ v3 \\ v4 \\ v5 \end{array} \begin{array}{c} \begin{array}{ccccc} v1 & v2 & v3 & v4 & v5 \end{array} \\ \begin{bmatrix} 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \end{bmatrix} \end{array}$$

Eg. Directed graph.



$$A = \begin{array}{c} \\ v1 \\ v2 \\ v3 \\ v4 \\ v5 \end{array} \begin{array}{c} \begin{array}{ccccc} v1 & v2 & v3 & v4 & v5 \end{array} \\ \begin{bmatrix} 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \end{array}$$

NOTE: for  2 edges or 3 edges we write accordingly into the matrix

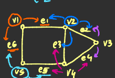
② Incidence Matrix

Let G be a graph with m vertices V1, V2, V3 .... Vm & n edges e1, e2, e3 ... en

let matrix $M = [m_{ij}]_{m \times n}$ defined by.

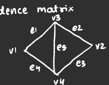
$$m \begin{cases} 1 & \text{if vertex } V_i \text{ is incident on } e_j \\ 0 & \text{if vertex } V_i \text{ is NOT incident on } e_j \\ 2 & \text{if vertex } V_i \text{ is self-loop.} \end{cases}$$

Eg.



$$A = \begin{array}{c} \\ v1 \\ v2 \\ v3 \\ v4 \\ v5 \end{array} \begin{array}{c} \begin{array}{cccccc} e1 & e2 & e3 & e4 & e5 & e6 \end{array} \\ \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix} \end{array}$$

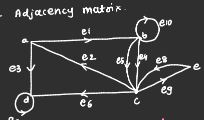
Q.1 Incidence matrix



→

$$A = \begin{array}{c} \\ v1 \\ v2 \\ v3 \\ v4 \end{array} \begin{array}{c} \begin{array}{ccccc} e1 & e2 & e3 & e4 & e5 \end{array} \\ \begin{bmatrix} 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 \end{bmatrix} \end{array}$$

Q.2 Adjacency matrix.



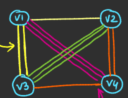
→

$$A = \begin{array}{c} \\ a \\ b \\ c \\ d \\ e \end{array} \begin{array}{c} \begin{array}{ccccc} a & b & c & d & e \end{array} \\ \begin{bmatrix} 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 2 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} \end{array}$$

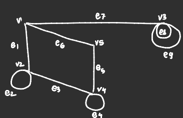
Q.3 Construct graph from full adjacency matrix

$$A = \begin{array}{c} \\ v1 \\ v2 \\ v3 \\ v4 \end{array} \begin{array}{c} \begin{array}{cccc} v1 & v2 & v3 & v4 \end{array} \\ \begin{bmatrix} 0 & 1 & 2 & 3 \\ 1 & 0 & 3 & 2 \\ 2 & 3 & 0 & 1 \\ 3 & 2 & 1 & 0 \end{bmatrix} \end{array}$$

→



Q.4 Adjacency & incidency of.



$$A: \begin{array}{c} \\ v1 \\ v2 \\ v3 \\ v4 \\ v5 \end{array} \begin{array}{c} \begin{array}{ccccc} v1 & v2 & v3 & v4 & v5 \end{array} \\ \begin{bmatrix} 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 2 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 \end{bmatrix} \end{array}$$

$$M = \begin{array}{c} \\ v1 \\ v2 \\ v3 \\ v4 \\ v5 \end{array} \begin{array}{c} \begin{array}{ccccccccc} e1 & e2 & e3 & e4 & e5 & e6 & e7 & e8 & e9 \end{array} \\ \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 2 & 2 \\ 0 & 0 & 1 & 2 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \end{bmatrix} \end{array}$$

we count self loop as 2 in incidence matrix.

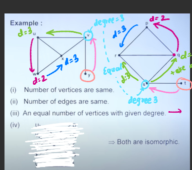
ISOMORPHISM & HOMEOMORPHISM

ISOMORPHISM

Two graphs G1 and G2 are said to be isomorphic if there is one to one correspondence between the edge set E1 and E2 in such a way that if e1 is an edge with end vertices u1 and v1 (in G1) then corresponding edge e2 in G2 has its end point vertices u2 and v2 (in G2) which corresponds to u1 and v1

Another definition:
Two graphs are isomorphic if,

① No. of vertices are same
② No of edges are same
③ An equal number of vertices with given degree
④ Vertex correspondence & edge correspondence valid



G1: u → 3, v → 3, w → 2, x → 3, y → 1

G2: p → 2, q → 3, r → 3, s → 3, t → 1

Checking one one correspondence...

- y — t } (∵ same degree & unique)
- w — p
- v — ? as there are 3 vertices of degree 3 ! in G2

so,

- v — s
- u — q
- w — p
- x — r

Vertex correspondence ✓

Edge correspondence,

if these r connected: y — t, w — p ← then these should also be connected for

- v — s, u — q
- w — p, x — r

[: connected

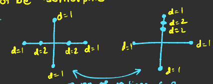
[: not connected

Homeomorphic Graph

Two graphs G and G' are said to be homeomorphic if they can be obtained from the same graph

NOTE: G & G' are homeomorphic, they need not be isomorphic.

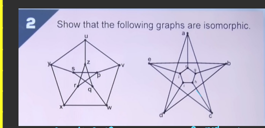
Eg.



① no. of vertices = 6
② no of edges = 4] Homeomorphic ✓
③ d=1 × 4 & d=2 × 2 both places.

Both are Homeomorphic coz both obtained from

Conditions
→ 1) No. of vertices & edges should be : same
2) Sequence of degree of vertices : same
3) Mapping of vertices not same.



1) No of vertices = 10	1) No of vert. = 10
2) ~~–~~ edges = 15	2) edges = 15
3) all have degree deg = 3	3) all have degree deg = 3

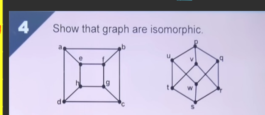
4) Mapping (vertices correspond.)

u — a	s — i
y — d	q — j
x — b	z — f
w — e	r — g
v — c	
p — h	

(p वरून s ला गेलो)
(h वरून कुठे जाऊ g वर की i वर?)
Ans: s येतो p & y च्या मध्ये
we know p — h & y — d
तर toh vertice h & d च्या मध्ये पाहिजे, जो की "i" आहे!

Edges corresp.
all are corresponding correctly
Eg. u & y connected — a & d connected.
⋮

Hence Proved, they are Isomorphic graphs.



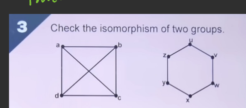
vert = 8	vert = 8
edges = 12	edges : 12
d = 3	d = 3

start with

- a — p
- b — q
- c — w
- d — u
- h — t (a & h च्या मध्ये म्हणजे)
- e — v ← (p & t च्या मध्ये)
- f — r
- g — s

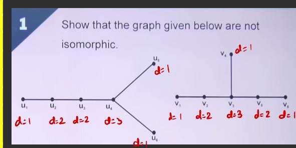
all connected, so,

Proved. Isomorphic graphs.



no of vert = 4 ≠ no of vert = 6

NOT isomorphic



vert = 6	← same →	vert = 6
edge = 5	← same →	edge = 5
[d=3] × 1, [d=2] × 2, [d=1] × 3	← same →	[d=3] × 1, [d=2] × 2, [d=1] × 3

Vertices correspondence.

Let,

- u1 — v1 (d=1, d=1)
- u2 — v2) पण v2 वरून v3 ला जाताना v3 ची d=3 आहे! # ERROR
- u3 — v3

u2 वरून u3 ला गेलो u3 ची d=2 होती

If we take

- u1 — v6
- u2 — v5
- u3 — v3 ← same error

If we take

- u1 — v4
- u2 — v3 ← Error! (d=2, d=3)

⇒ Hence, not Isomorphic

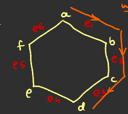
(istri kapde)

WALK, TRAIL, PATH

① Walk

A walk is a finite alternating sequence of vertices and edges, beginning and ending with same or diff. vertices.

Eg.



walk: $ae_1be_2ce_3d$

Starting from a — Ending at d

Length of walk
The number of edges is called as length of the walk

Closed and open walk
If its origin and terminous vertex ($V_0 = V_n$) is equal then is called Closed walk, otherwise open walk
└ if diff vertices ($V_0 \neq V_n$)

TRAIL

Any walk having all different edges is called as train
or no repeated edges.

CIRCUIT

A closed trail is called as a circuit.

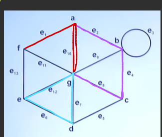
PATH

A walk is called path if all vertices are not repeated

CYCLE

A closed path is called as cycle

(Eg.)



$ae_2be_4ce_8g$
↳ Path ✓
Circuit ✗
Trail ✓
Cycle ✗

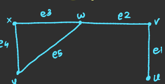
$ge_7de_6ee_{12}g$ → Path ✓
Cycle ✓
Trail ✓
Circuit ✓

$fe_1ae_{10}ge_{10}a$
↳ walk ✓
Path ✗ as a is repeated
Trail ✗ as e_{10} repeated.

NOTE: This repetition is allowed
closed loop आहे so allowed आहे.

EULERIAN PATH

A path in a graph is said to be Eulerian Path if it traverses each & every edge in graph once & only once



then $ue_1ve_2we_3xe_4ye_5w$
is Eulerian.

{NOTE: Vertice repetition is accepted here.}

EULERIAN CIRCUIT

A circuit in a graph is said to be an Eulerian circuit if it traverses each edge in graph once & only once.
(for closed graph)

HAMILTONIAN PATH

A path which contain every vertex of a graph G exactly once is called Hamiltonian graph.

Eg.

$ve_1ue_6we_5ye_4x$
Hamiltonian path

HAMILTONIAN CIRCUIT

A circuit that passes through each of the vertices in a group G exactly once except the starting vertex and end vertex is called Hamiltonian circuit

Eg.1

→ $ue_1ve_2we_3xe_4ye_5u$
Hamiltonian ckt

Eg.2 Above diag.

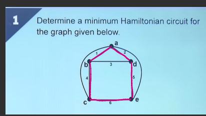
on joining this line, vertex w repeats which is not allowed so, NOT Hamiltonian Circuit!

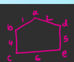
Hamiltonian Graph:

A connected graph containing Hamiltonian circuit is called as Hamiltonian graph.

✓ Hamiltonian graph

✗ Not Hamiltonian graph.

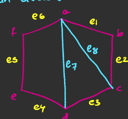




⇒ a2d5e6c4b1a minimum Hamiltonian ckt.

Q.2 Draw a graph with 6 vertices containing a Hamiltonian circuit but not Eulerian circuit.

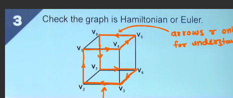
→



$ae_1be_2ce_3de_4ee_5fe_6a$ — Hamiltonian Ckt
all vertices not repeated
but become Eulerian ckt also as # all edges not repeated

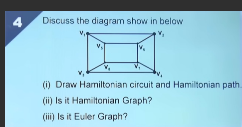
For Eulerian ckt, added e_7 & e_8 ←
$ae_1be_2ce_3de_4ee_5fe_6ae_7de_3ce_8a$
Error e_3 repeated

so now, its just Hamiltonian circuit & not Eulerian circuit

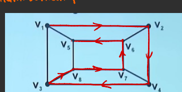


Hamiltonian circuit in a graph
∴ Hamiltonian graph ✓

Cannot be Eulerian graph as some or other edge misses out / repeats

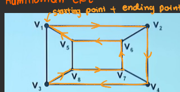


Hamiltonian path



Hamiltonian ckt

starting point + ending point



Hamiltonian ckt present in graph
so ✓ Hamiltonian Graph

Eulerian Graph (trials) for Eulerian ckt.

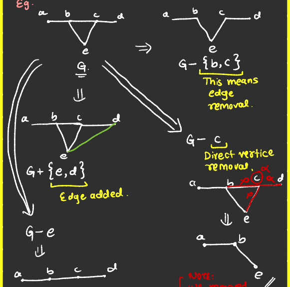
so not Eulerian Graph!

Operations on Graph

Removing vertex from Graph:

When we remove a vertex v and all edges incident/connected to it from Graph { G = (V, E) } we create a subgraph, denoted by G − v (minus)

Eg.



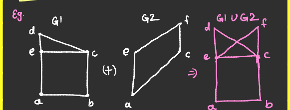
Graph Union

Two or more graphs can be combined in various ways. The new graph that contains all the vertices and edges of these graphs is called as union of graph.

The union of two simple graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ is simple graph with vertex set $V_1 \cup V_2$ and edge set $E_1 \cup E_2$

The union of G_1 and G_2 is denoted by $G_1 \cup G_2$

Eg.

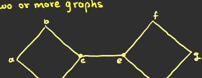


Cut Vertex & Cut Set

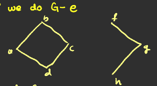
Connectivity:
A graph is said to be connected if there is path between every pair of vertex.
From one vertex to any other vertex, such that there should be atleast some path to traverse.

Cut Vertex
A vertex $v \in G$ is called as Cut-Vertex of G if G − v (G minus v) results in a disconnected graph.
Removing a cut vertex from a graph breaks it into two or more graphs.

Eg.



Here 'e' & 'c' are Cut vertices as if we do G − e



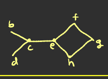
disconnected

& if we do G − c

disconnected.

If we remove any other vertices except c & e the graph is not gonna break / disconnect

Eg. G − a



connected only!

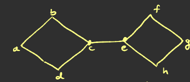
Applications of Hamilton Circuits

Hamilton paths and circuits can be used to solve practical problems. For example, many applications ask for a path or circuit that visits each road intersection in a city, each place pipelines intersect in a utility grid, or each node in a communications network exactly once. Finding a Hamilton path or circuit in the appropriate graph model can solve such problems. The famous **traveling salesperson problem** or TSP (also known in older literature as the **traveling salesman problem**) asks for the shortest route a traveling salesperson should take to visit a set of cities. This problem reduces to finding a Hamilton circuit in a complete graph such that the total weight of its edges is as small as possible. We will return to this question in Section 10.6.

Cut-Edge (Bridge)

An edge $e \in G$ is called as a cut edge, if 'G − e' results into a disconnected graph.
If removing an edge results into two/more graph then that edge is called as a cut edge.

Eg.



⇒ here Cut edge is {(c, e)}

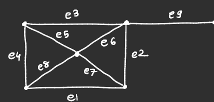
NOTE 'G' graph with 'n' vertices
1. Cut edge e exist iff it is not a part of cycle!
2. Maximum number of cut edges possible is 'n-1'.
3. Whenever cut edges exist, cut vertices also exists.
4. Whenever cut vertices exist, cut edge may or may not exist.

Cut Set of a Graph

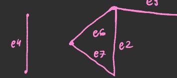
Let Graph be G = (V, E) be a connected graph. A subset E' of E, is called cut-set of G if deletion of all edges given in E' set causes G to disconnect

→ If deleting certain number of edges from a graph makes it disconnected, then those deleted edges are called as cut set (set of) of graph

Eg

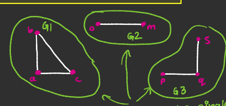


Cut set is $E_1 = \{e1, e3, e5, e8\}$
(# remove all these edges)



disconnected graph!

CONNECTED COMPONENTS



These can be single graph also.

Suppose in Qn had it been given draw graph with 8 vertices & 6 edges so that
a — b
b — c
a — c
o — m
p — q
q — s
} then we would have drawn this graph 'G'

So, the graph G is the union of 3 disjoint connected subgraphs G1, G2 & G3
These subgraphs are called as Connected components of G.

CONNECTED COMPONENTS A **connected component** of a graph *G* is a connected subgraph of *G* that is not a proper subgraph of another connected subgraph of *G*. That is, a connected component of a graph *G* is a maximal connected subgraph of *G*. A graph *G* that is not connected has two or more connected components that are disjoint and have *G* as their union.

APPLICATIONS

APPLICATIONS OF PLANAR GRAPHS Planarity of graphs plays an important role in the design of electronic circuits. We can model a circuit with a graph by representing components of the circuit by vertices and connections between them by edges. We can print a circuit on a single board with no connections crossing if the graph representing the circuit is planar. When this graph is not planar, we must turn to more expensive options. For example, we can partition the vertices in the graph representing the circuit into planar subgraphs. We then construct the circuit using multiple layers. (See the preamble to Exercise 30 to learn about the thickness of a graph.) We can construct the circuit using insulated wires whenever connections cross. In this case, drawing the graph with the fewest possible crossings is important. (See the preamble to Exercise 26 to learn about the crossing number of a graph.)

The planarity of graphs is also useful in the design of road networks. Suppose we want to connect a group of cities by roads. We can model a road network connecting these cities using a simple graph with vertices representing the cities and edges representing the highways connecting them. We can built this road network without using underpasses or overpasses if the resulting graph is planar.