UNIT-V

Para

GRAPH THEORY

Introduction: Graph theory is intimately related to many branches of mathematics. It is widely applied in subjects like computer technology, communication science, electrical engineering, Architecture, physics, operations Research etc.

Graph theory was born in 1736, with Euler's paper on Konigsberg bridge problem. Konigsberg bridge is the best known example in graph theory. It was a long pending problem. Euler solved this problem by means of a graph. Euler become father of graph theory. Kirchoff, Mobius, Demorgan and Hamilton have laid strong Foundations and contributed much to the development of the subject.

Scanned by TapScanner

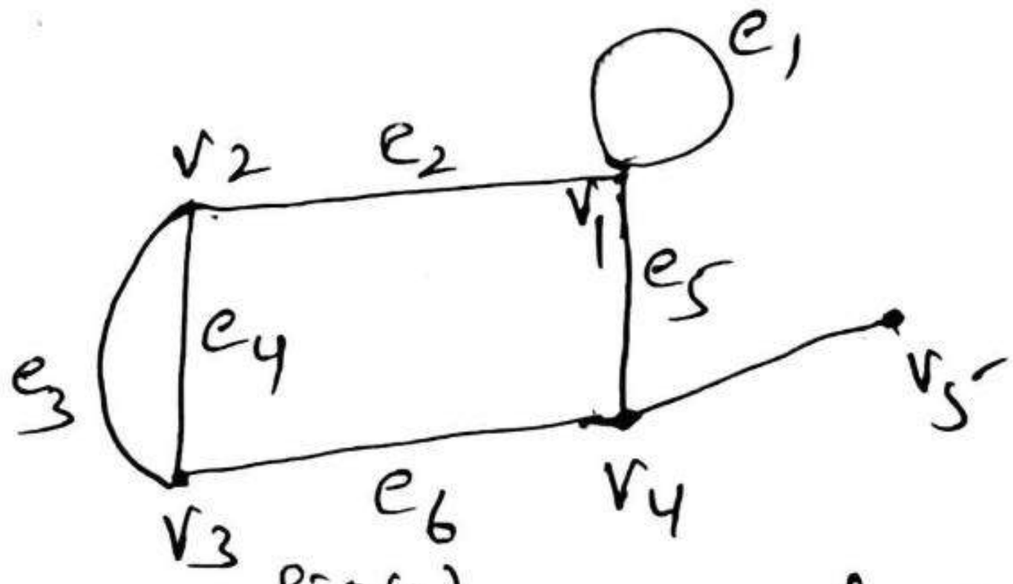
BASIC CONCEPTS

Graph: A graph $G=(V,E)$ consists of two $\{v_1, v_2, \ldots\}$ objects V and E such that $V=\{v_1, v_2, \ldots v_k\}$ called vertices and $E=\{e_1, e_2, \ldots e_n\}$ $\{e_1, e_2, \ldots\}$ called edges and each edge e_k is associated with an unordered pair (v_i, v_j) of vertices.

Note: 1) Graphs are represented by means of diagrams.

2) If an edge e_k is associated with (v_i, v_j) then v_i and v_j are called the ends of e_k.

Ex: 1)



fig(1)

Graph with 5 vertices and 7 edges.

Loop: In a graph G, if an edge e_k has both of its ends as v_i, i.e, e_k is associated with (v_i, v_i) then e_k is called a loop.

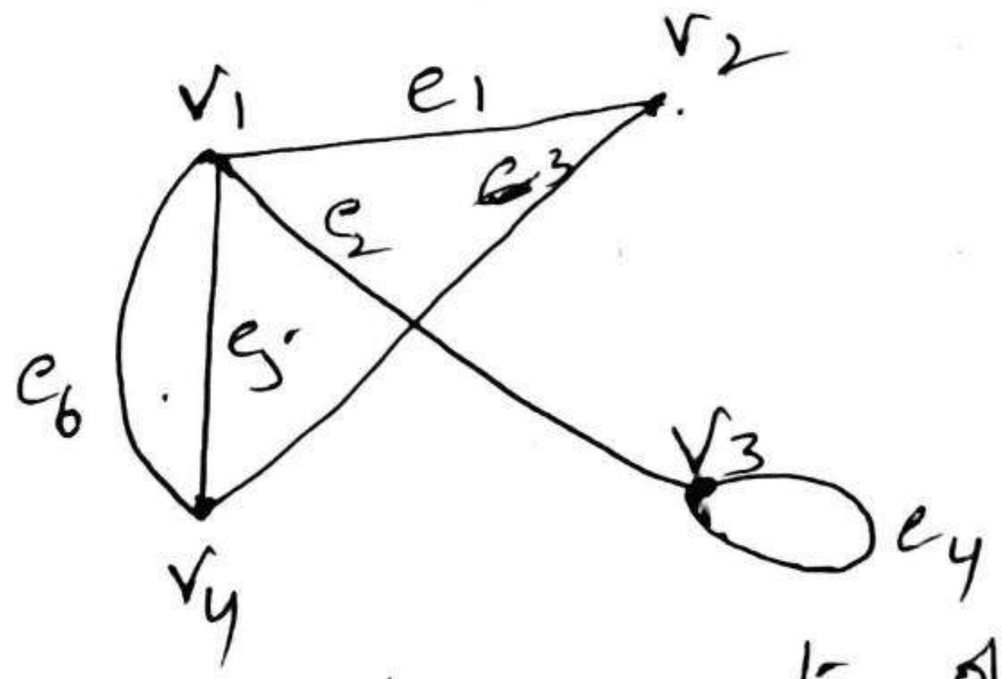
Ex: In fig(1), e_1 is a loop, since it is associated with (v_1, v_1)

Parallel edges or multiple edges

In a graph G, if there are more than one edges associated with the same pair of vertices then they are called parallel edges or multiple edges.

Ex: In the above example (1), e_3 and e_4 are parallel edges, since they are associated with the single pair (v_2, v_3).

NO consider the graph shown in figure. we note



that this graph consists of four vertices v_1, v_2, v_3, v_4 and six edges $e_1, e_2, e_3, e_4, e_5, e_6$. Although the edges e_2 and e_3 seem to intersect (cross over) in the figure, their point of intersection (even

when it exists) is not a vertex of the graph.

We observe that the edges e_1, e_2, e_3 have distinct end vertices, but the edge e_4 has the same vertex v_3 as both of its end vertices i.e, $e_4 = (v_3, v_3)$

An edge such as e_4 is called <u>a loop</u>.

We also observe that both of the edges e_5 and e_6 have the same end vertices v_1, v_4;

i.e $e_5 = (v_1, v_4)$ and $e_6 = (v_1, v_4)$.

Edges such as these are called <u>Parallel edges</u>.

If in a graph there are two or more edges with the same end vertices, the edges are called <u>multiple edges</u>.

<u>Order and size</u>:

The number of vertices in a (finite) graph is called the order of the graph and the number of edges in it is called its size.

In other words, for a graph $G = (V, E)$ the cardinality of the set V, namely $|V|$,

is called the order of G and the cardinality of the set E, namely $|E|$, is called the size of G. A graph of order n and size m is called a (n, m) graph. ~~Thus the graph depicted~~ in fig. The

A null graph with n vertices is a (n, 0) graph

→ Definition of a Directed graph:

A directed graph (or a digraph) is a pair (V, E), where V is a nonempty set and E is a set of ordered pairs of elements taken from the set V.

For a directed graph (V, E), the elements of V are called vertices (points or nodes) and the elements of E are called directed edges. The set V is called the vertex set and the set E is called the directed edge set.

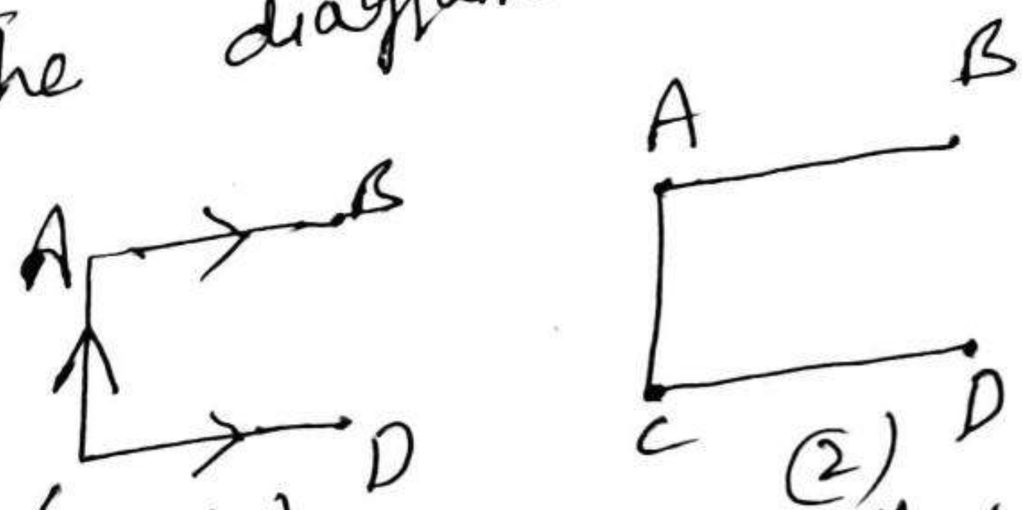
Scanned by TapScanner

In-degree and out-degree :

If v is a vertex of a digraph D, the no. of edges for which v is the initial vertex is called the out-going degree or the out-degree of v and the number of edges for which v is the terminal vertex is called the incoming degree or the in-degree of v.

The out-degree of v is denoted by $d^+(v)$ or $od(v)$ and the in-degree of v is denoted by $d^-(v)$ or $id(v)$.

Graphs: Let us draw the diagram by dropping the arrows present in the edges. The diagram appears as shown below.



We say that this diagram represents the diagram of an underlying graph of the

First thm of the Digraph Theory: (After in-degree out degree) (3a)

Property X

In every digraph D, the sum of the out-degrees of all vertices is equal to the sum of the in-degrees of all vertices, each sum being equal to the number of edges in D.

Pf: Suppose D has n vertices $v_1, v_2, \cdots v_n$ and m edges.

Let r_1 be the no. of edges going out of v_1.

r_2 be the no. of edges going out of v_2

and so on.

Then

$$d^+(v_1) = r_1, \quad d^+(v_2) = r_2 \cdots \quad d^+(v_n) = r_n.$$

Since every edge terminates at some vertex and since there are m edges, we should have

$$r_1 + r_2 + \cdots + r_n = m$$

Accordingly,

$$d^+(v_1) + d^+(v_2) + \cdots + d^+(v_n) = r_1 + r_2 + \quad + r_n = m$$

Similarly,

If s_1 is the number of edges coming into v_1, s_2 is the number of edges coming into v_2, and so on

we get

$$d^-(v_1) + d^-(v_2) + \cdots + d^-(v_n) = s_1 + s_2 + \cdots + s_n = m$$

Thus,

$$\sum_{i=1}^{n} d^+(v_i) = \sum_{i=1}^{n} d^-(v_i) = m.$$

This completes the proof.

Scanned by TapScanner

directed graph shown in fig (1).
This underlying graph has the same vertices and the same edges as the original directed graph, but its edges are all undirected.

Definition of a Graph: A Graph is a pair (V, E), where V is a nonempty set and E is a set of unordered pairs of elements taken from the set V.

For a graph (V, E), the elements of V are called vertices (or points or nodes) and the elements of E are called undirected edges or just edges. The set V is called the vertex set and the set E is called the edge set.

The graph (V, E) is also denoted by $G = (V, E)$ or $G = G(V, E)$. or just G

Null graph: A graph / digraph containing no edges is called a null graph.

→ A null graph with only one vertex is called a trivial graph.

Scanned by TapScanner

fig shows a null graph with three vertices.

→ In diagrams depicting non-null graphs, an edge is represented by a line. The line may be straight or curved, long or short.

<u>Note</u>: If G is a directed graph (digraph), the elements of E are ordered pairs of vertices. In this case an edge (u, v) is said to be from u to v, and to ~~joined~~ join u to v.

If G is a nondirected graph the elements of E are unordered pairs of vertices. In this case an edge {u, v} is said to join 'u' and 'v' or to be between 'u' and v.

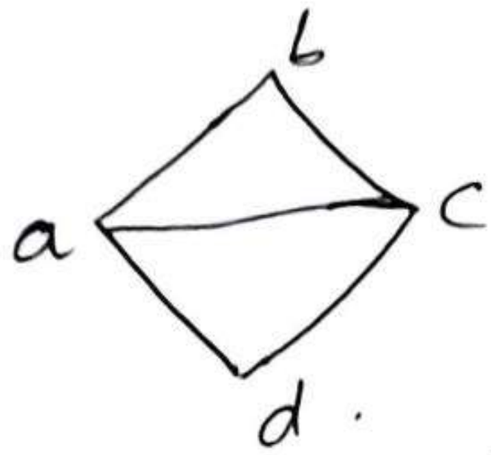
→ <u>Finite Graph</u> / digraph: A graph / digraph with only a finite number of vertices as well as only a finite no. of edges is called a finite graph / digraph. otherwise it is called an infinite graph / digraph.

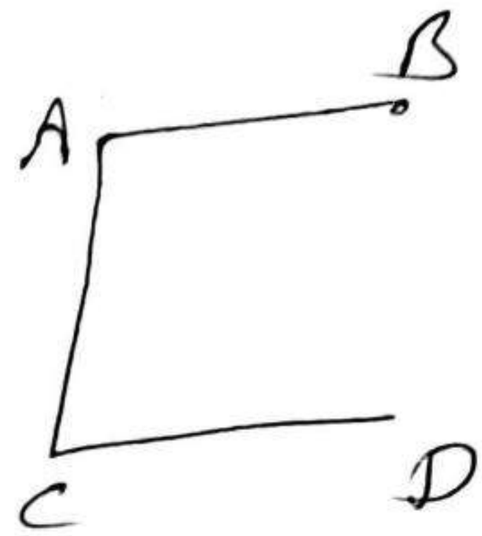
Scanned by TapScanner

Types of Graphs:

Simple graph: A graph which does not contain loops and multiple edges is called a simple graph. A graph which does not contain loop is called a loop-free graph.

Ex:

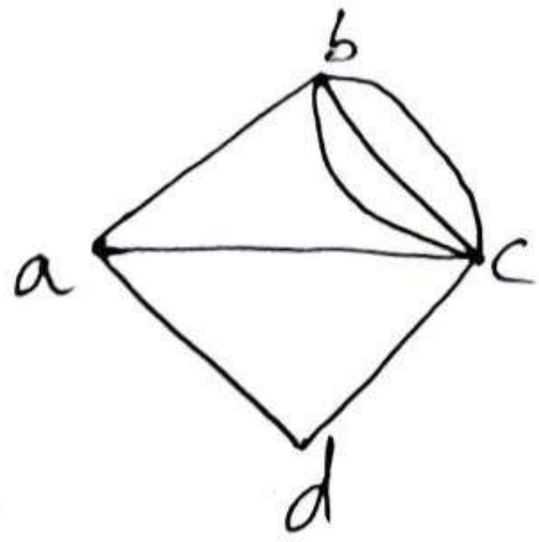




simple graph

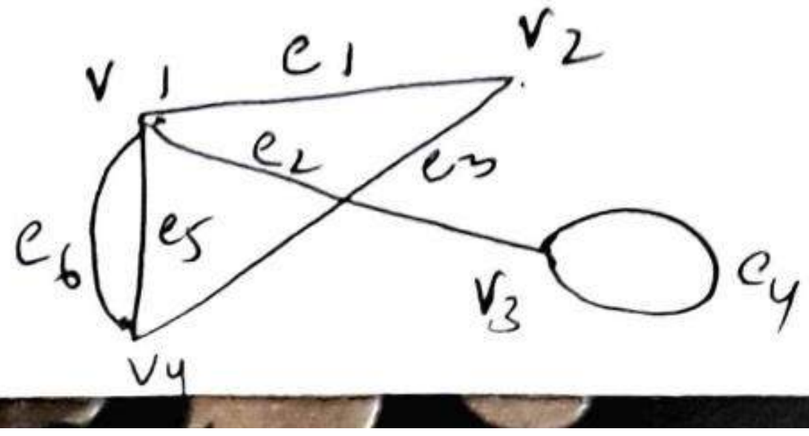
Multi graph: A graph which contains multiple edges but no loops is called a multigraph.

Ex:



General graph:

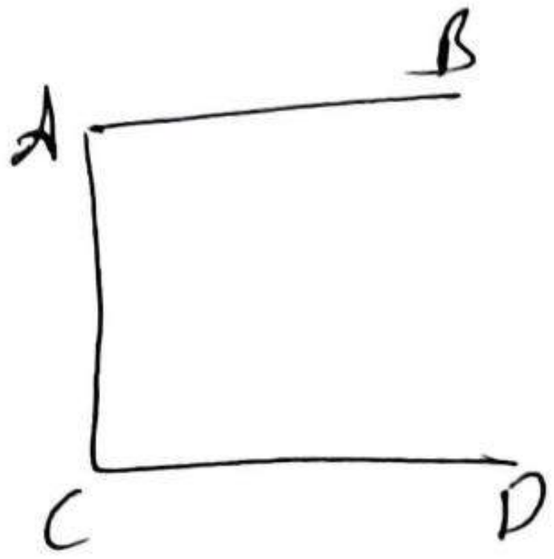
A graph which contains multiple edges or loops (or both) is called a general graph.



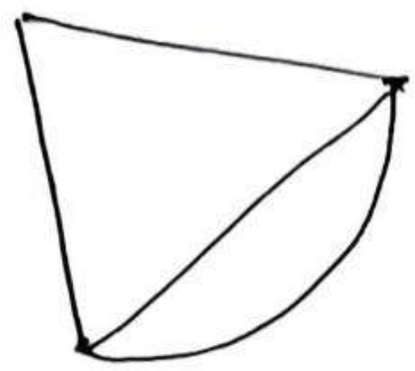
Scanned by TapScanner

Labeled graph and unlabeled graph:

In many situations, the names assigned to vertices are inconsequential. In such situations we do not assign any names to the vertices. Such graph are called unlabeled graph. On the other hand, if names are assigned to vertices of a graph, the graph is called a labeled graph.



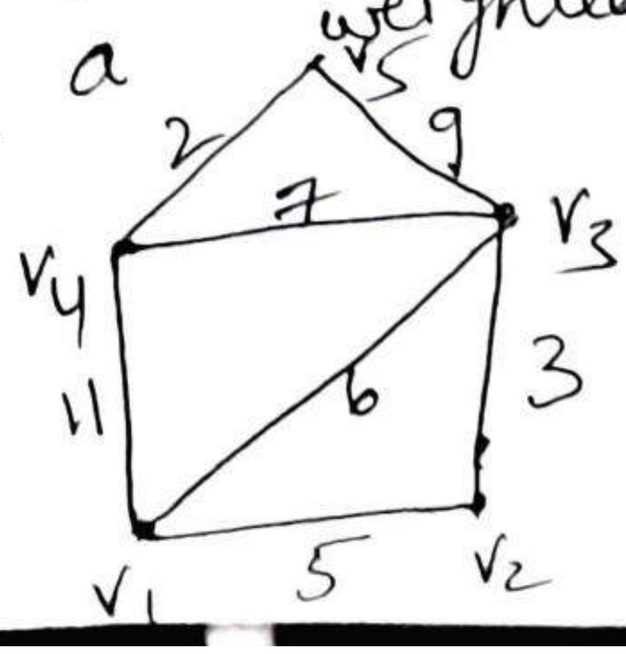
labeled graph



unlabeled graph.

Incidence weighted Graph:

A Graph G is in which weights are assigned to every edge is called a weighted graph.



Incidence (between vertices and edges)

Let G be a graph, if $e = \{u, v\}$, then the edge 'e' is said to be incident with the vertices u and v.

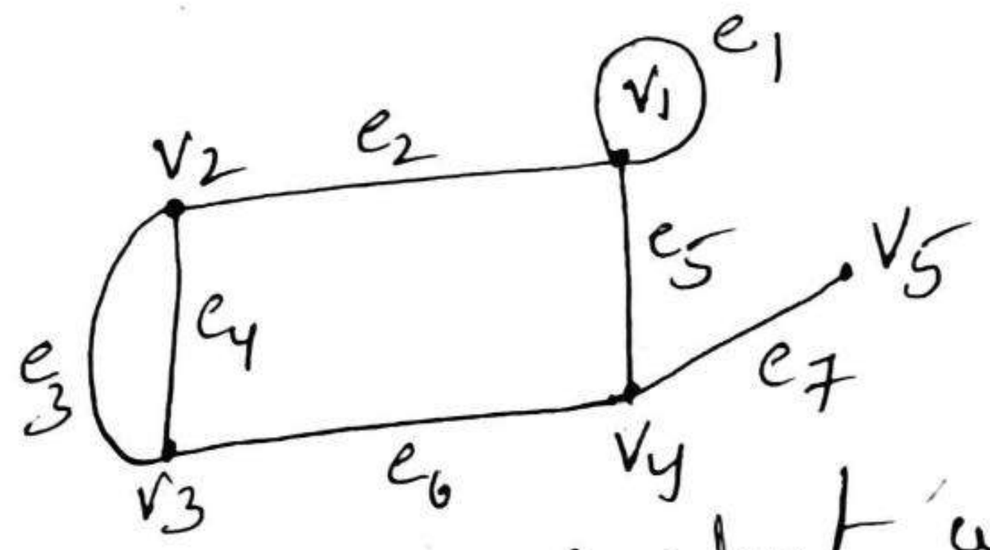
The edge 'e' is also said to connect u and v, and u, v are called the ends of e.

$$u \overset{e}{\text{———}} v$$

Adjacency (between two vertices)

Let G be an undirected graph, two vertices 'u' and v are said to be adjacent (or neighbors) in G if there is an edge b/w 'u' and v.

Ex:



Incidence: e_2 is incident with v_1, v_2

e_3 and e_4 are incident with v_2, v_3

e_1 is incident with v_1 only.

Adjacent: v_4 and v_5 are adjacent.

v_2 and v_3 are "

But v_2 and v_4 are not adjacent.

Scanned by TapScanner

Degree (of a vertex): Let $G=(V,E)$ be an undirected graph. The degree of a vertex v in G is the no. of edges incident on v, with a loop at v contributes twice to the degree of that vertex.

The degree of a vertex is denoted by $\deg(v)$.

→ The degrees of the vertices of a graph arranged in non-decreasing order is called the degree sequence of the graph. Also, the minimum of the degrees of vertices of a graph is called the degree of the graph.

Ex: In the above figure, the degrees of vertices of the graph are

$\deg(V_1)=4$, $\deg(V_2)=3$, $\deg(V_3)=3$

$\deg(V_4)=3$, $\deg(V_5)=1$. Therefore, the degree sequence of the graph is 1, 3, 3, 3, 4 and the degree of the graph is 1.

Adjacency of vertices in directed graphs:

Let $G=(V,E)$ be a directed graph. If (u,v) is an edge of G then u is said to be adjacent to v and v is said to be adjacent from u.

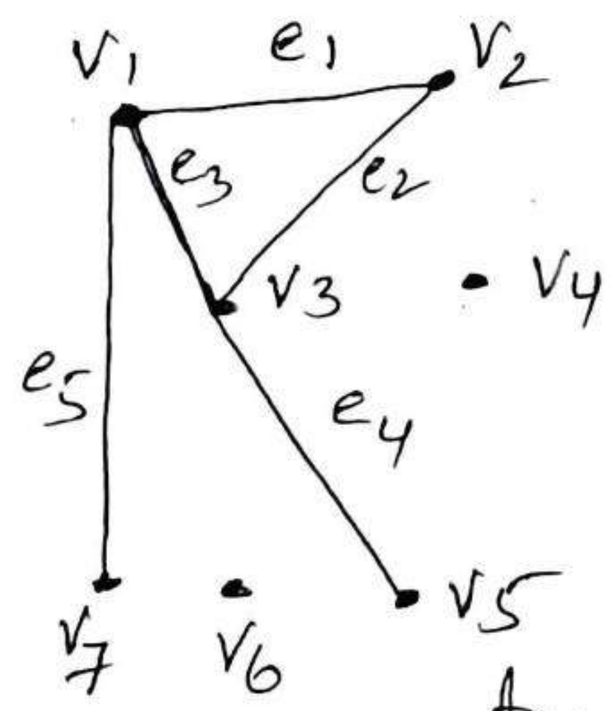
Note: 1) If (u,v) is an edge in directed graph $G=(V,E)$, then the vertex u is called the initial vertex of (u,v) and v is called the terminal or end vertex of (u,v).

2) If the edge is a loop then the initial and terminal vertices are the same.

Isolated vertex, pendant vertex

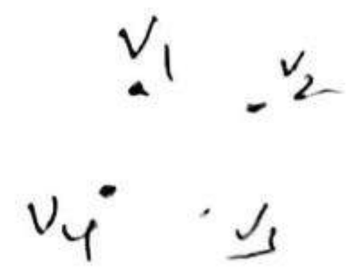
A vertex in a graph which is not an end vertex of any edge of the graph is called an isolated vertex. Obviously, a vertex is an isolated vertex if and only if its degree is zero.

→ A vertex of degree 1 is called a pendant vertex. An edge incident on a pendant vertex is called a pendant edge.



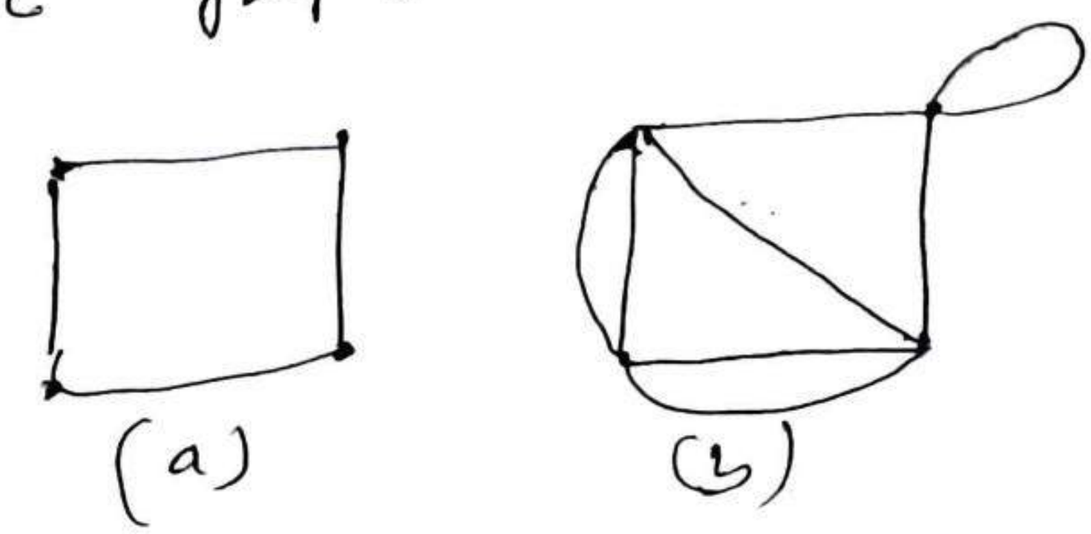
In the graph, the vertices v_4 and v_6 are isolated vertices, v_5 and v_7 are pendant vertices and the edges e_4 and e_5 are pendant edges.

Note: A null graph contains no edges. It follows that in a null graph every vertex is an isolated vertex.



Regular graph: A graph in which all the vertices are of the same degree 'K' is called a regular graph of degree K, or a K-regular graph.

→ In particular, 3-regular graphs are called cubic graphs.



(a) (b)

The graph (a) is 2-regular graph and the graph (b) is 4-regular graph.

Handshaking Property: or The First thm of graph theory

Thm The sum of the degrees of all the vertices in a graph is an even number, and this number is equal to twice the number of edges in the graph.

OR) (i) The sum of degree of the vertices of a nondirected graph $G=(V,E)$ is twice the number of edges in G.

For a graph $G=(V,E)$, we have

$$\sum \deg(v) = 2|E|.$$

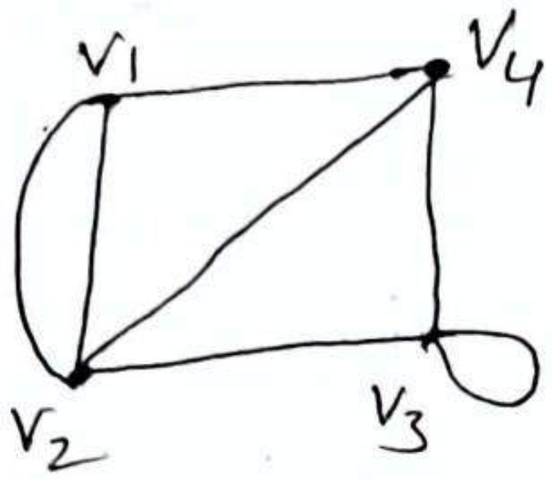
i.e $\sum_{i=1}^{n} \deg(v_i) = 2|E|$

(ii) If G is a directed graph, then

$$\sum_{i=1}^{n} \deg^+(v_i) = \sum_{i=1}^{n} \deg^-(v_i)$$

Scanned by TapScanner

Ex:



$\deg(v_1) = 3$, $\deg(v_2) = 4$, $\deg(v_3) = 4$, $\deg(v_4) = 3$.

Also, the graph has 7 edges.

So $\deg(v_1) + \deg(v_2) + \deg(v_3) + \deg(v_4) = 14 = 2 \times E$
$= 2 \times 7$.

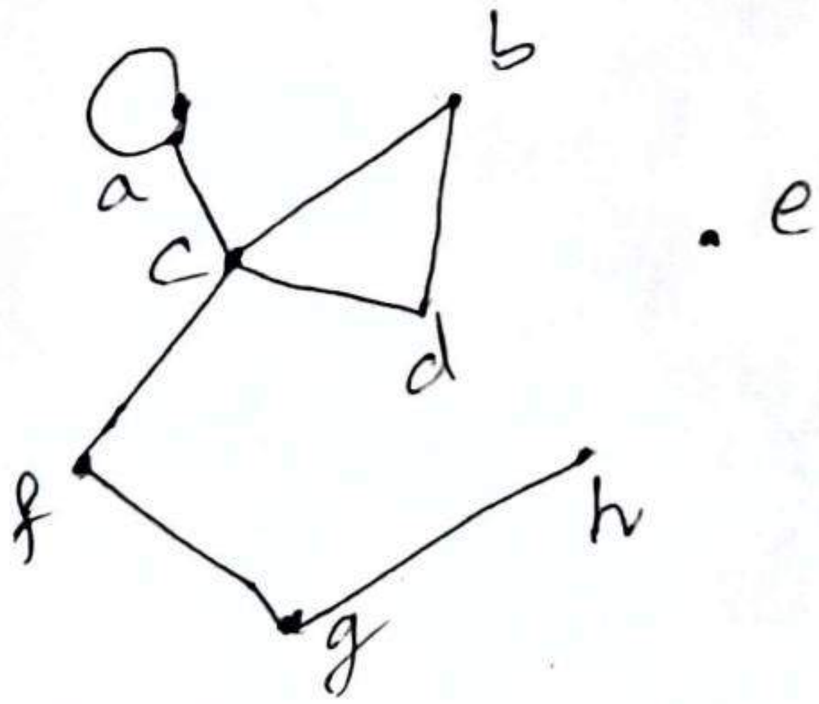
<u>Thm</u> In every graph, the no. of vertices of odd degree is even. $\deg(v_1) = 3$, $\deg(v_2) = 4$, $\deg(v_3) = 4$, $\deg(v_4) = 3$

Ex: In the above graph, the degree of the graph is 3. which is odd.

So, the number of vertices are $= 4$ i.e even.

Problem

<u>Ex 1.)</u> For the graph show below, indicate the degree of each vertex and verify the handshaking property.



By examining the graph, we find that the degree of its vertices are

$\deg(a) = 3$, $\deg(b) = 2$, $\deg(c) = 4$

$\deg(d) = 2$ $\deg(e) = 0$, $\deg(f) = 2$, $\deg(g) = 2$

$\deg(h) = 1$

we note that e is an isolated vertex and h is a pendant vertex.

Also, the sum of the degrees of vertices is

$\deg(a) + \deg(b) + \dots + \deg(h) = 16$

Also, the graph has 8 edges i.e $E = 8$.

$\therefore \deg(a) + \deg(b) + \dots + \deg(h) = 2 \times 8$.

Thus, the sum of the degrees of vertices is equal to twice the number of edges.

$\therefore$ The handshaking property is verified

⇒ 2) How many edges does a graph ∴(C) have if it has vertices of degree 4, 3, 3, 2, 2? Draw such a graph.

<u>Sol</u>: Let G be a graph with vertices degrees 4, 3, 3, 2, 2.

Since sum of the degrees of the vertices is twice to its number of edges.

i.e $\sum d(v) = 2e$, where 'e' denotes the no. of edges.

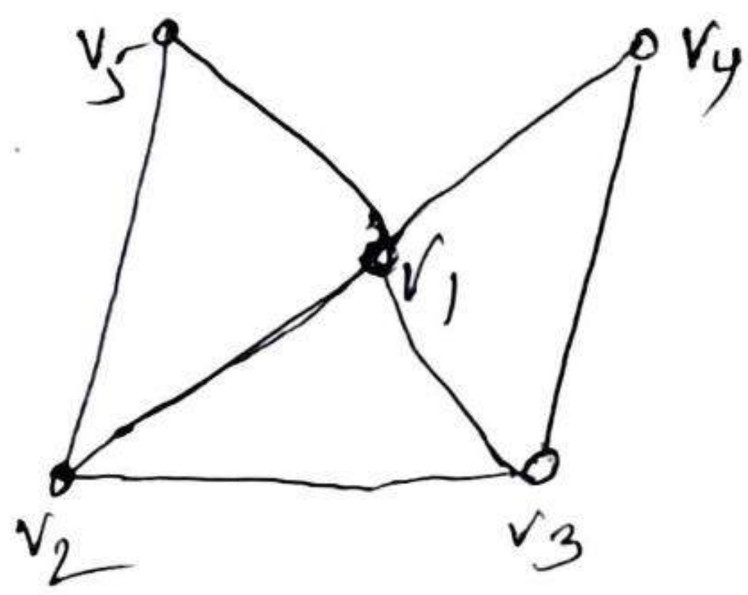
By the hypothesis, we have

$$4+3+3+2+2 = 2e.$$

$$\Rightarrow 14 = 2e$$

$$\Rightarrow e = 7.$$

So, the number of edges of graph G with its vertices degree 4, 3, 3, 2, 2 is 7.



In this graph $d(v_1) = 4$, $d(v_2) = 3$, $d(v_3) = 3$, $d(v_4) = 2$, $d(v_5) = 2$.

Scanned by TapScanner

How many vertices will the graph contains 6 '6' edges and all vertices of degree 3.

Sol: Given that graph G contains 6 edges and all vertices of G are of degree 3.

Now, we have to find the no. of vertices of graph G.

Since the sum of the degrees of the vertices of a graph G is twice to its number of edges.

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

$\Rightarrow \Sigma d(v) = 2 \times 6 = 12$

Since each vertex degree 3 and sum of the degrees of the vertices is 12 gives that $(3+3+3+3 = 12)$ graph G contains four vertices.

Ans: How many vertices will be in a graph containing 16 edges and all vertices of degree 2.

Ans: The graph G contains 16 vertices with each vertex of degree 2.

Complete graph: The complete graph on 'n' vertices, denoted by K_n, is the simple graph that contains exactly one edge b/w each pair of distinct vertices.

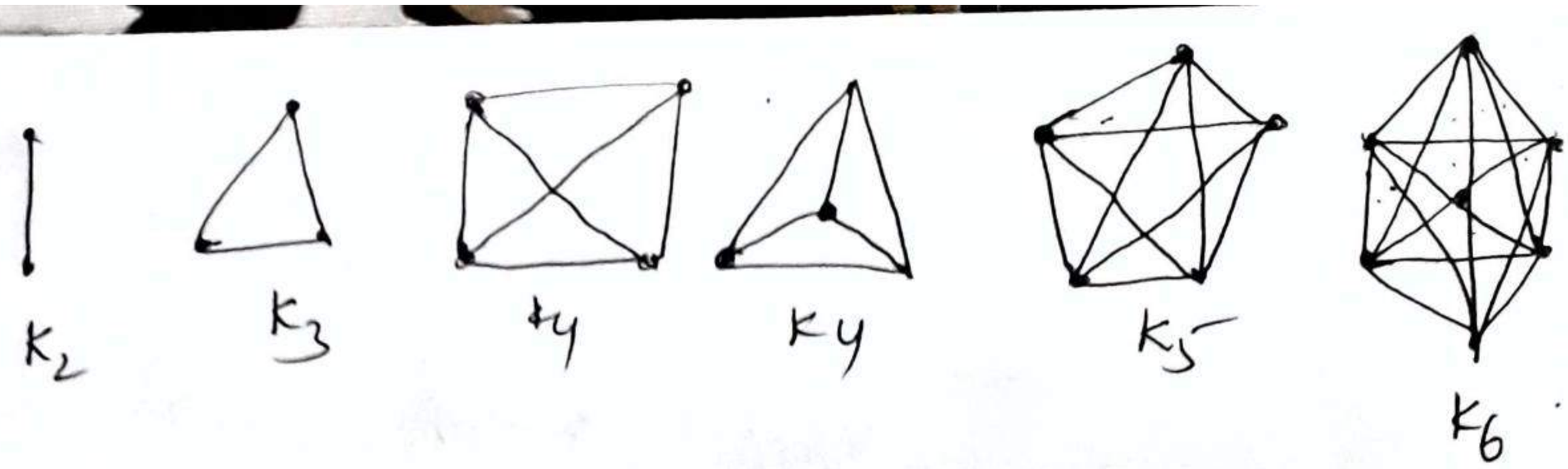
Def^n: A simple graph of order ≥ 2 in which there is an edge b/w every pair of vertices is called a complete graph (or a full graph).

(In otherwords, a complete graph is a simple graph in which every pair of distinct vertices are adjacent)

A complete graph with $n \geq 2$ vertices is denoted by K_n.

complete graph with two, three, four and five vertices are shown in figure. Of these complete graphs, the complete graph with five vertices, names K_5 is of great importance. This graph is called the kuratowski's first graph.

Scanned by TapScanner



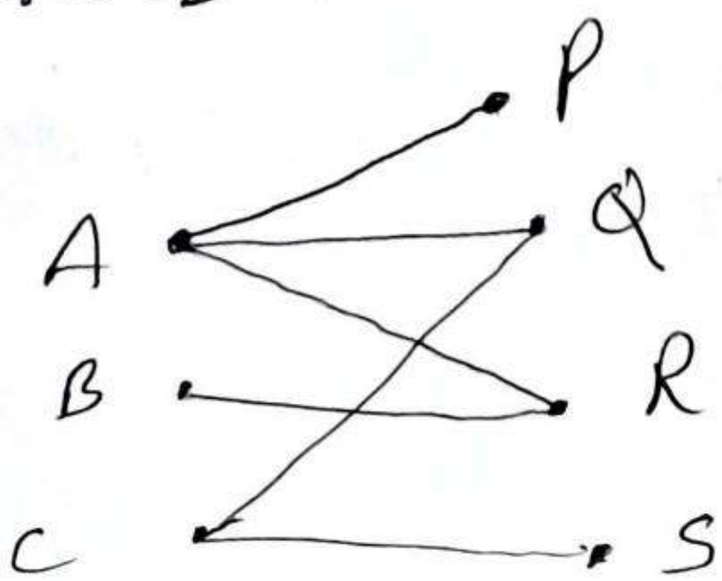
Bipartite graph:

Suppose a simple graph G is such that its vertex set V is the union of two mutually disjoint nonempty sets V_1 and V_2 which are such that every edge in G joins a vertex in V_1 and a vertex in V_2. Then G is called a bipartite graph.

If E is the edge set of this graph, the graph is denoted by $G = (V_1, V_2; E)$

The sets V_1 and V_2 are called bipartites (or partitions) of the vertex set V.

Ex: consider the graph G shown in fig.



for which the vertex set is ~~V = {A, B, C, P, Q, R}~~

$V = \{A, B, C, P, Q, R, S\}$ and the edge set is

$E = \{AP, AQ, AR, BR, CQ, CS\}$.

Note that the set V is the union of two of its subsets $V_1 = \{A, B, C\}$ and $V_2 = \{P, Q, R, S\}$ which are such that (i) V_1 and V_2 are disjoint (ii) every edge in G is a join of a vertex in V_1 and a vertex in V_2.

(iii) G contains no edge that joins two vertices both of which are in V_1 or V_2.

This graph is a bipartite graph with $V_1 = \{A, B, C\}$ and $V_2 = \{P, Q, R, S\}$ as bipartite

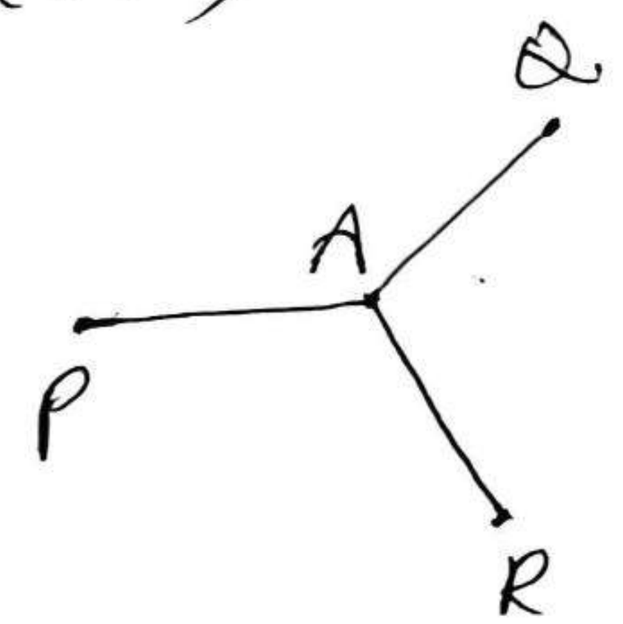
Complete Bipartite graph:

A bipartite graph $G = (V_1, V_2; E)$ is called a complete bipartite graph if there is an edge b/w every vertex in V_1 and every vertex in V_2.

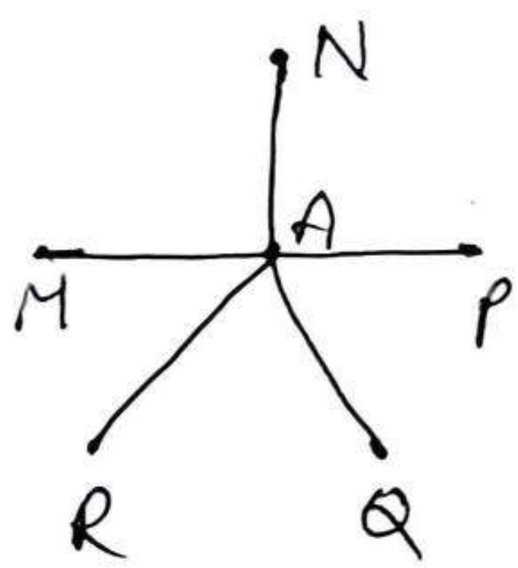
The bipartite graph shown above is not a complete bipartite graph. for

example, that the graph does not contain an edge joining A and S.

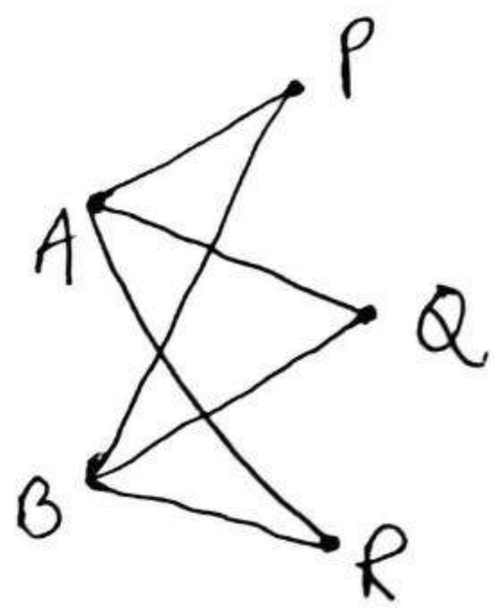
A complete bipartite graph $G=(V_1, V_2, E)$ in which the bipartites V_1 and V_2 contain r and s vertices respectively, with $r \le s$, is denoted by $K_{r,s}$. In this graph, each of r vertices in V_1 is joined to each of s vertices in V_2. Thus $K_{r,s}$ has $r+s$ vertices and rs edges, i.e, $K_{r,s}$ is of order $r+s$ and size rs; it is therefore a $(r+s, rs)$ graph.



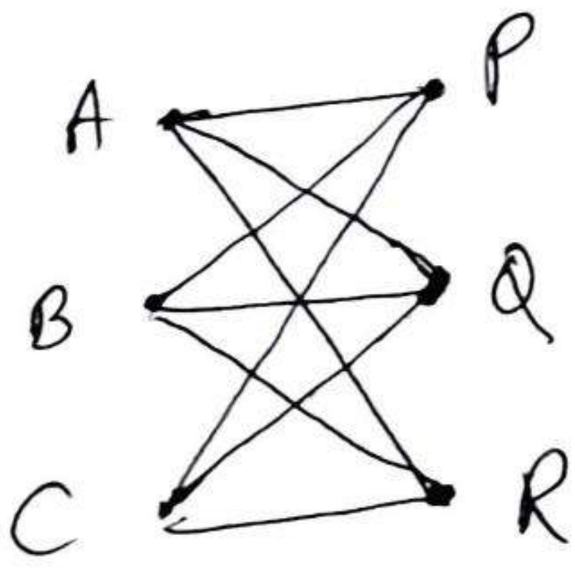
(a) $K_{1,3}$



(b) $K_{1,5}$



(c) $K_{2,3}$



(d) $K_{3,3}$

figures (a) to (d) depict some complete bipartite graphs. observe that in figure (a), the bipartites are $V_1=\{A\}$ and $V_2=\{P, Q, R\}$, the vertex 'A' is joined to each of the vertices P, Q, R

by an edge. In fig(b), the bipartites are $V_1 = \{A\}$, and $V_2 = \{M, N, P, Q, R\}$, the vertex A is joined to each of the vertices M, N, P, Q, R by an edge.

In fig(c), the Bipartites are $V_1 = \{A, B\}$ and $V_2 = \{P, Q, R\}$, each of the vertices A and B is joined to each of the vertices P, Q, R by an edge.

In fig(d), the bipartites are $V_1 = \{A, B, C\}$ and $V_2 = \{P, Q, R\}$, each of the vertices A, B, C is joined to each of the vertices P, Q, R.

→ Of these complete bipartite graphs, the graph $K_{3,3}$ shown in fig (d) is of great importance. This is known as the Kuratowski's second graph.

<u>Note</u>: It is to be noted that a bipartite Graph G is not a complete graph even if G is a complete bipartite graph. Because in such a graph, there exists no edge b/w two vertices if they belong to the same bipartite.

Problems

1) Draw a diagram of the graph $G=(V,E)$ in each of the following cases:

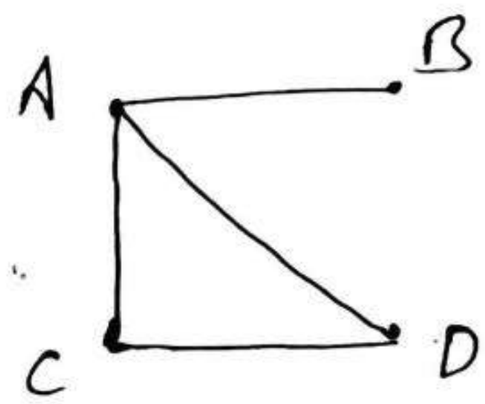
(i) $V=\{A,B,C,D\}$ $E=\{(A,B),(A,C),(A,D),(C,D)\}$

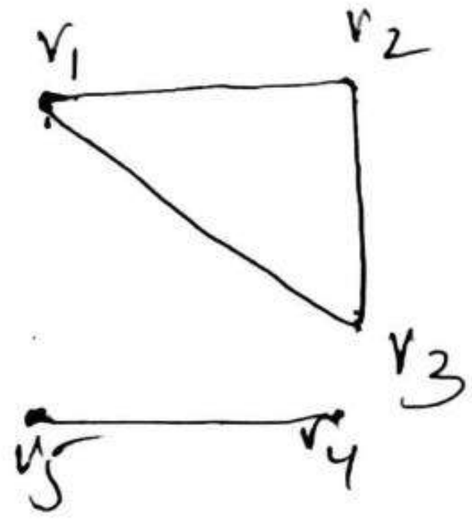
(ii) $V=\{v_1,v_2,v_3,v_4,v_5\}$, $E=\{(v_1,v_2),(v_1,v_3),(v_2,v_3),(v_4,v_5)\}$

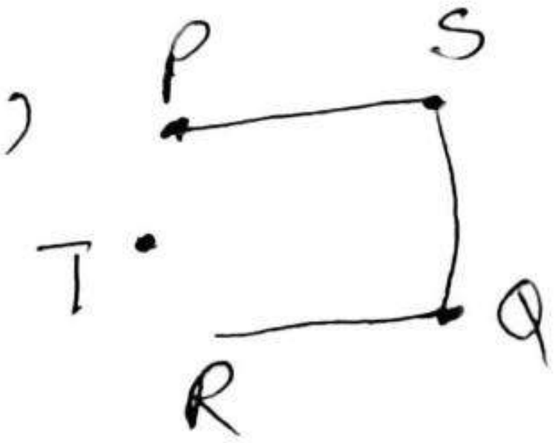
(iii) $V=\{P,Q,R,S,T\}$, $E=\{(P,S),(Q,R),(Q,S)\}$

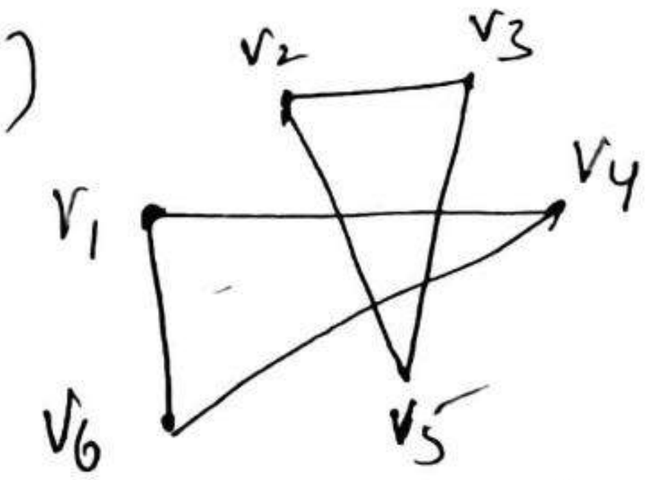
(iv) $V=\{v_1,v_2,v_3,v_4,v_5,v_6\}$, $E=\{(v_1,v_4),(v_1,v_6),(v_4,v_6),(v_3,v_2),(v_3,v_5),(v_2,v_5)\}$

Sol: The required diagrams are

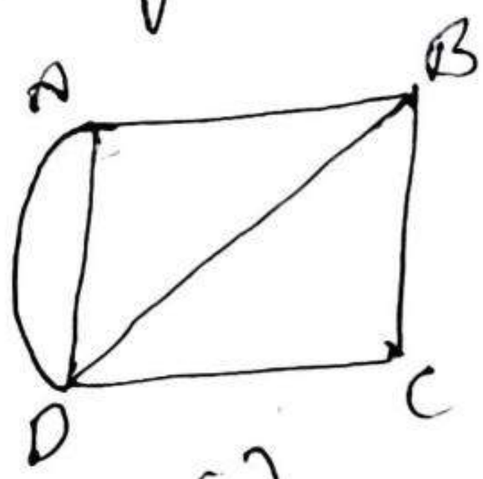
(i)


(ii)


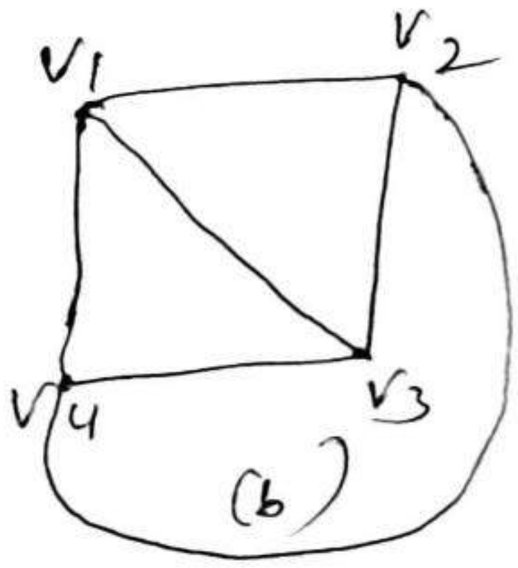
(iii)


(iv)


2) Which of the following are complete graphs.



(a)

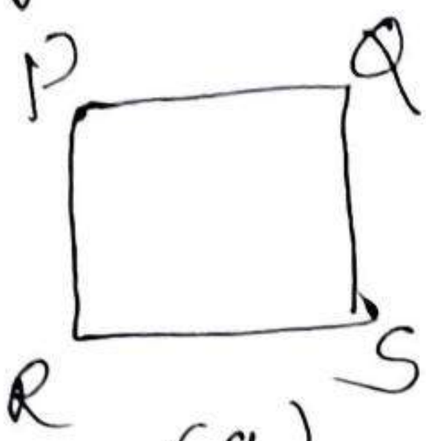


(b)

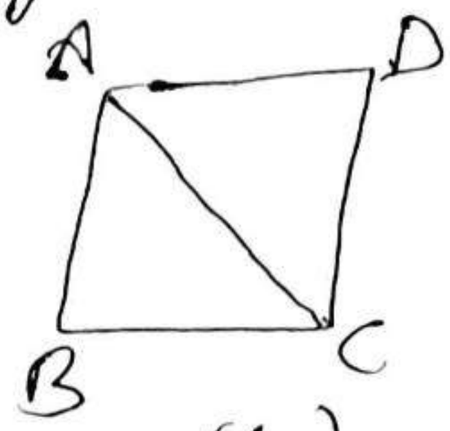
Sol: The first of the graphs is not complete. It is

not simple on the one hand and there is no edge b/w A and C on the other hand. The second of the graphs is complete. It is simple and there is an edge b/w every two vertices.

3) which of the following is a bipartite graph?



(a)



(b)

$v_1 = \{P, S\}$, $v_2 = \{Q, R\}$

$v_1 = \{P, S\}$

$v_2 = \{$

Sol: The first of the graph (a) is a bipartite graph with $v_1 = \{P, S\}$ and $v_2 = \{Q, R\}$ of the bipartites. The second graph is <u>not a bipartite graph</u>

4) → (a) How many vertices and how many edges are there in complete bipartite graphs $K_{4,7}$ and $K_{7,11}$?

(b) if the graph $K_{r,12}$ has 72 edges, what is r?

Sol: The complete bipartite graph $K_{r,s}$ has $r+s$ vertices and rs edges. Accordingly:

(a) The graph $K_{4,7}$ has $4+7=11$ vertices and $4\times7=28$ edges, and the graph $K_{7,11}$ has 18

vertices and $11 \times 7 = 77$ edges.

(b) If the graph $K_{r,12}$ has 72 edges, then we have

$$12r = 72 \Rightarrow r = 6$$

Isomorphism:

Consider two graph $G = (V, E)$ and $G' = (V', E')$. suppose there exists a function $f : V \to V'$ such that (i) f is one-to one and onto.

(ii) For all vertices A, B of G, the edge $\{A, B\} \in E$ if and only if the edge $\{f(A), f(B)\} \in E'$.

Then f is called an isomorphism b/w G and G' and we say that G and G' are isomorphic graphs.

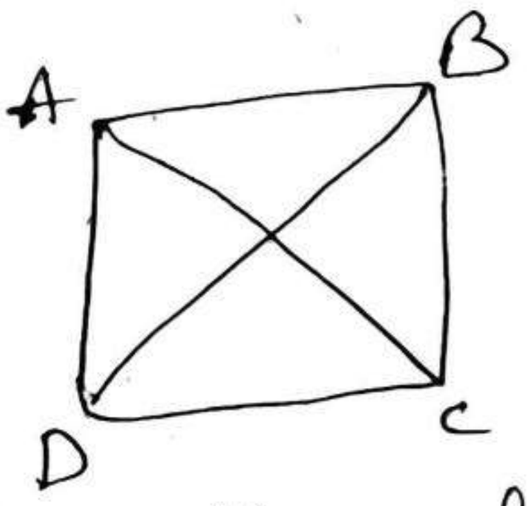
In other words, two graphs G and G' are said to be isomorphic (to each other) if there is a one-to-one correspondence b/w their vertices and b/w their edges such that the adjacency of vertices is preserved. Such graphs will

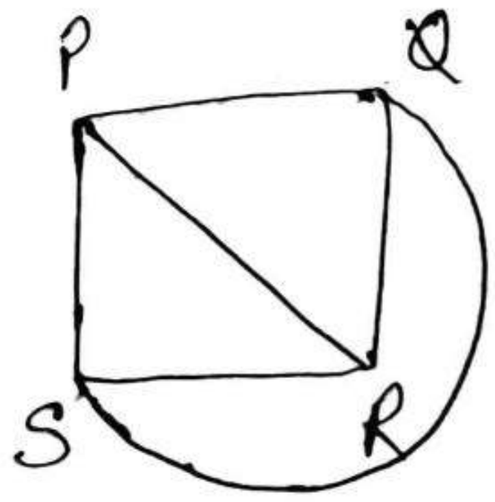
have the same structure, differing only in the way their vertices and edges are labeled or only in the way they are represented geometrically. For many purposes, we regard them as essentially the same.

When G and G' are isomorphic, we write $G \cong G'$.)

Ex: when a vertex A of G corresponds to the vertex $A' = f(A)$ of G' under an one-to-one correspondence $f: G \to G'$, we write $A \Leftrightarrow A'$. similarly, we write $\{A, B\} \Leftrightarrow \{A', B'\}$ to mean that the edge AB of G and the edge $A'B'$ of G' correspond to each other.

Ex:





consider the following one-one correspondence b/w the vertices of these two graphs

$A \Leftrightarrow P$, $B \Leftrightarrow Q$, $C \Leftrightarrow R$, $D \Leftrightarrow S$.

Scanned by TapScanner

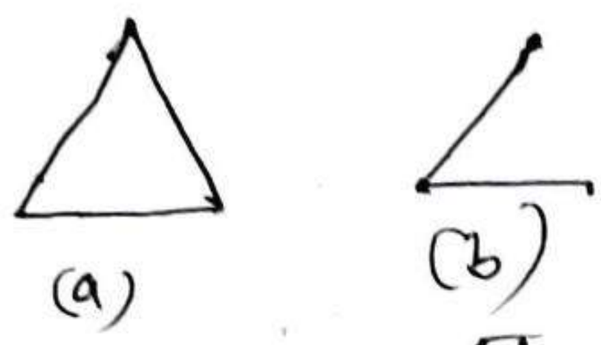
under this correspondence, the edges in the two graphs correspond with each other, as indicated below:

$\{A,B\} \leftrightarrow \{P,Q\}$, $\{A,C\} \leftrightarrow \{P,R\}$,

$\{A,D\} \leftrightarrow \{P,S\}$, $\{C,D\} \leftrightarrow \{R,S\}$, $\{B,C\} \leftrightarrow \{Q,R\}$

$\{B,D\} \leftrightarrow \{Q,S\}$

2) Ex: consider the graphs shown in figures (a) & (b)



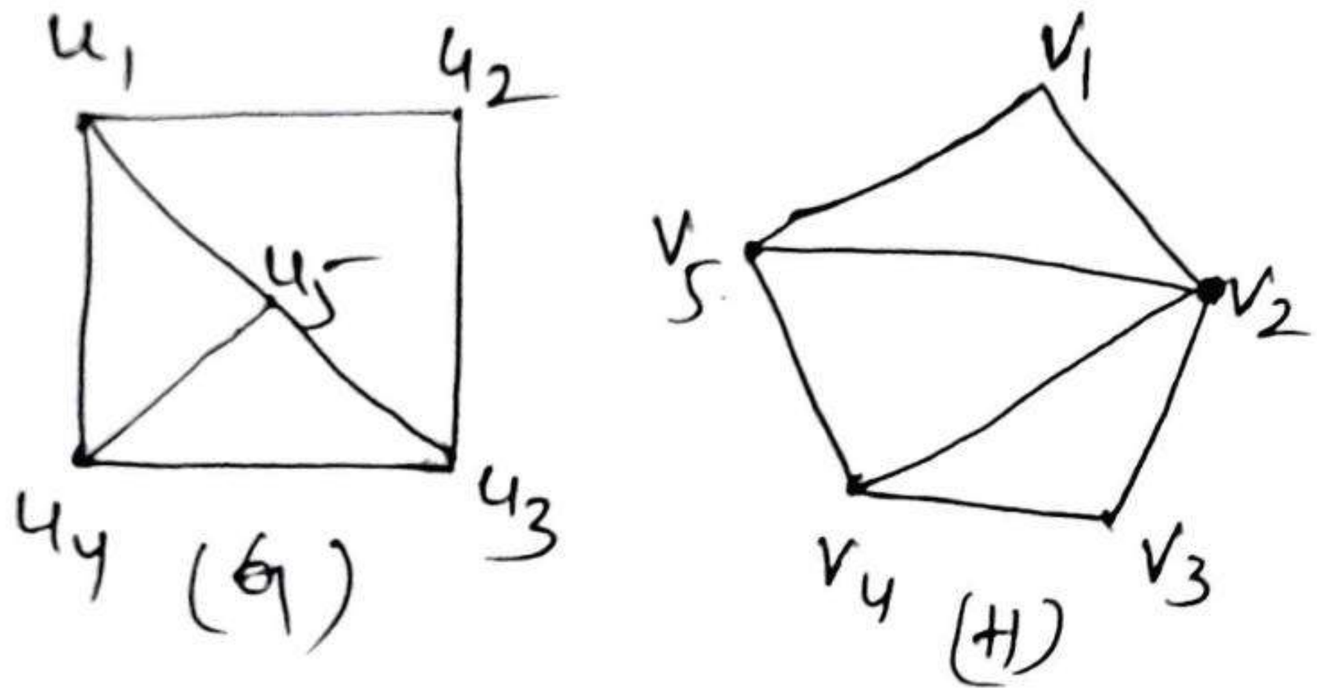
(a) (b)

Sol: we observe that the two graphs have the same number of vertices but different number of edges. Therefore, although there can exist one-to-one correspondence b/w the vertices, there cannot be a 1-1 correspondence b/w the edges. The two graphs are therefore not isomorphic.

→ from the def^n of isomorphism of graphs, it follows that if two graphs are isomorphic, then they must have:

1). The same number of vertices.
2). The same number of edges.
3). An equal number of vertices with a given degree.
4) The incident relationship must be preserved.

→ S.T the Graphs are isomorphic or not.



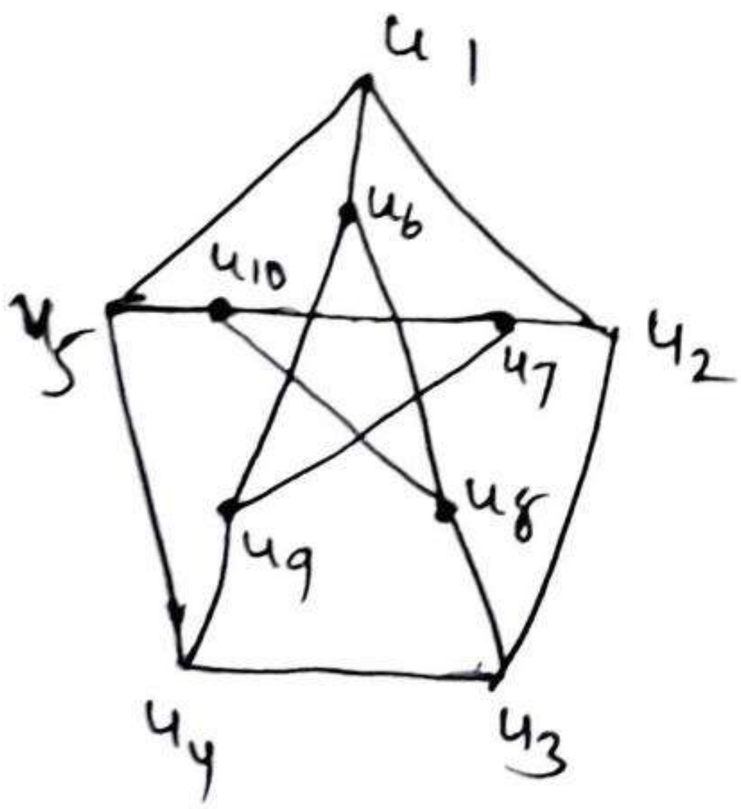
$u_1 = 3$ $v_1 = 2$
$u_2 = 2$ $v_2 = 4$
$u_3 = 3$ $v_3 = 2$
$u_4 = 3$ $v_4 = 3$
$u_5 = 3$ $v_5 = 3$

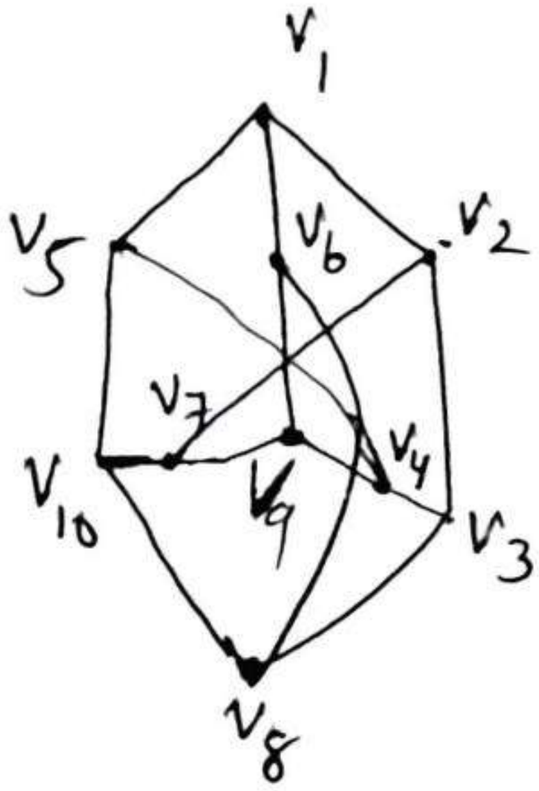
Sol: G has no vertex of degree 4.

But H has a vertex of degree 4.

Hence G & H are not isomorphic.

→





Sol: u_1 corresponds to v_1 u_6 corresponds to v_6

u_2 " v_2 u_7 " " v_7

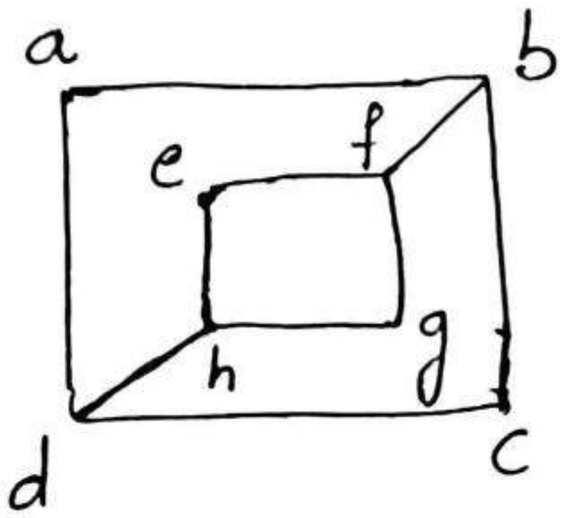
u_3 corresponds to v_3 u_8 " " v_8

u_4 corresponds to v_4 u_9 corresponds to v_9

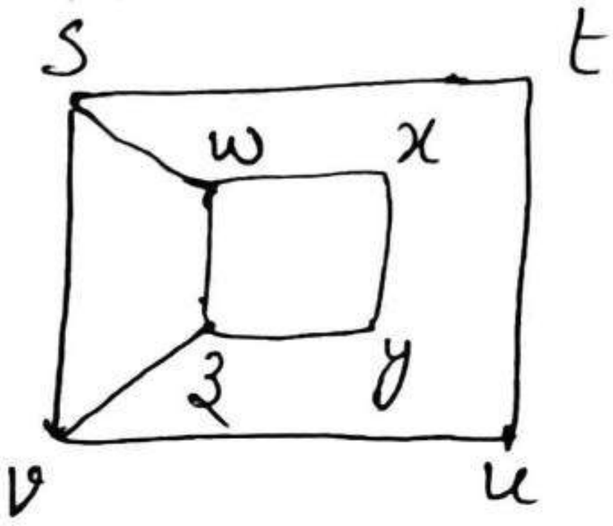
u_5 " " v_5 u_{10} " " v_{10}

∴ The graphs are isomorphic.

Ex:3). Determine whether the graphs G and H are isomorphic.



G



H

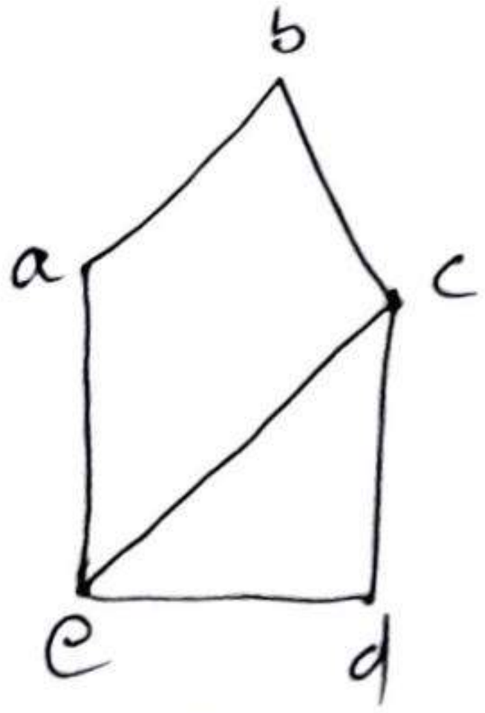
s = 3 w = 3
t = 2 x = 2
u = 2 y = 2
v = 3 z = 3

a = 2 ✓ e = 2 ✓
b = 3 f = 3
c = 2 ✓ g = 2 ✓
d = 3 h = 3 ✓

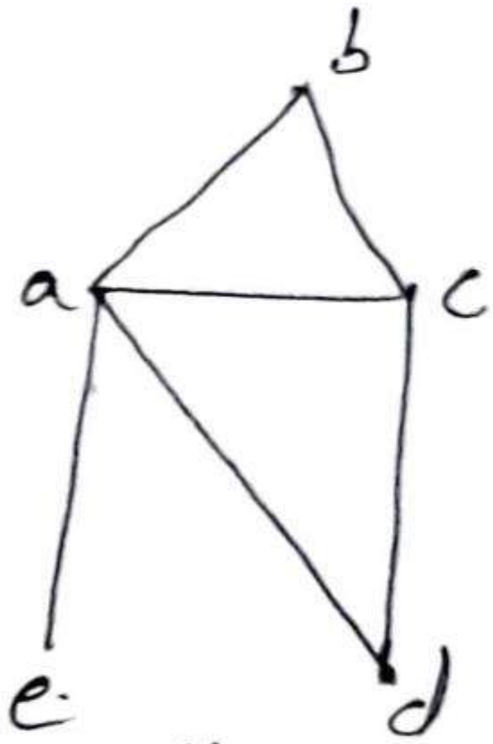
Sol: The graphs G and H both have 8 vertices and 10 edges. They also both have four vertices of degree 2 and four of degree 3. Since these invariants all agree it is still conceivable that these graphs are isomorphic.

However G and H are <u>not isomorphic</u> to see this note that since deg(a) = 2 in G 'a' must correspond to either t, u, x or y in H. Since these are the vertices of degree 2 in H. However each of these four vertices in H is adjacent to another vertex of degree 2 in H which is not true for 'a' in G.

4) S.T the graphs are not isomorphic.



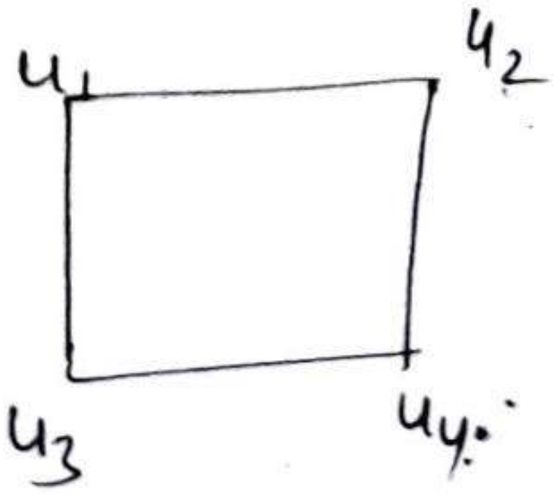
G

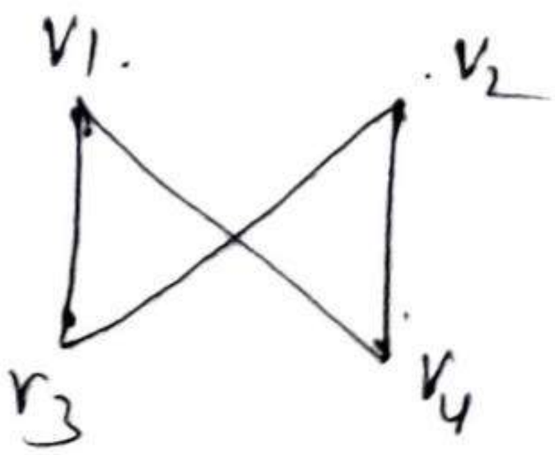


H.

Sol: Both G and H have five vertices and six edges. However H has a vertex of degree 1, namely e, whereas G has no vertices of degree 1. It follows that G and H are not isomorphic.

5) P.T the two graphs shown below are isomorphic





Sol: we first observe that both graphs have four vertices and four edges. consider the following one-to-one correspondence b/w the vertices of the graphs:

$u_1 \longleftrightarrow v_1$, $u_2 \longleftrightarrow v_4$, $u_3 \longleftrightarrow v_3$, $u_4 \longleftrightarrow v_2$

This correspondence gives the following correspondence

the edges:

$\{u_1, u_2\} \longleftrightarrow \{v_1, v_4\}$, $\{u_1, u_3\} \longleftrightarrow \{v_1, v_3\}$, $\{u_2, u_4\} \longleftrightarrow \{v_2, v_4\}$

$\{u_3, u_4\} \longleftrightarrow \{v_3, v_2\}$

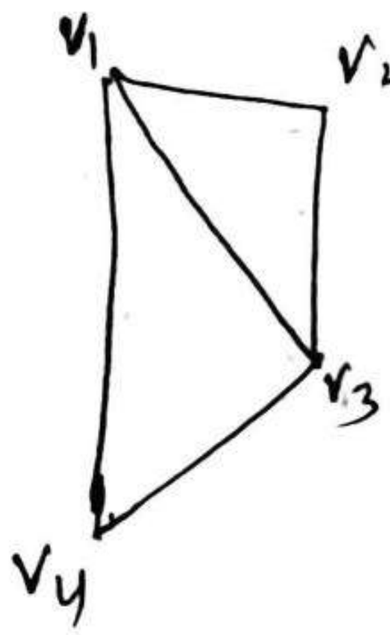
Subgraphs:

Given two graphs G and H, we say that 'H' is a subgraph of G if the following conditions hold:

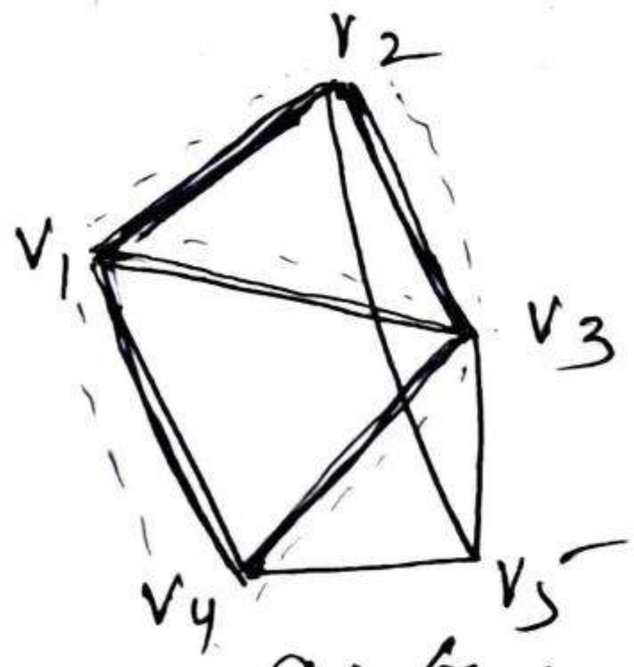
(1) All the vertices and all the edges of H are in G.

(2) Each edge of H has the same end vertices in G as in H.

→ consider the two graphs H and G shown in figures (a) and (b) respectively.



(a) H



(b) G

we observe that all vertices and all edges

of the graph H are in the graph G and that every edge in H has the same end vertices in G as in H.

Therefore, H is a subgraph of G. In the diagram of G, the part H is shown in thick lines

The following results are immediate consequences of the def^n of a subgraph:

1) Every graph is a subgraph of itself.

2) Every simple graph of n vertices is a subgraph of the complete graph K_n.

3) If H is a subgraph of a graph K and K is a subgraph of a graph G, then H is a subgraph of G.

4) A single vertex in a graph G is a subgraph of G.

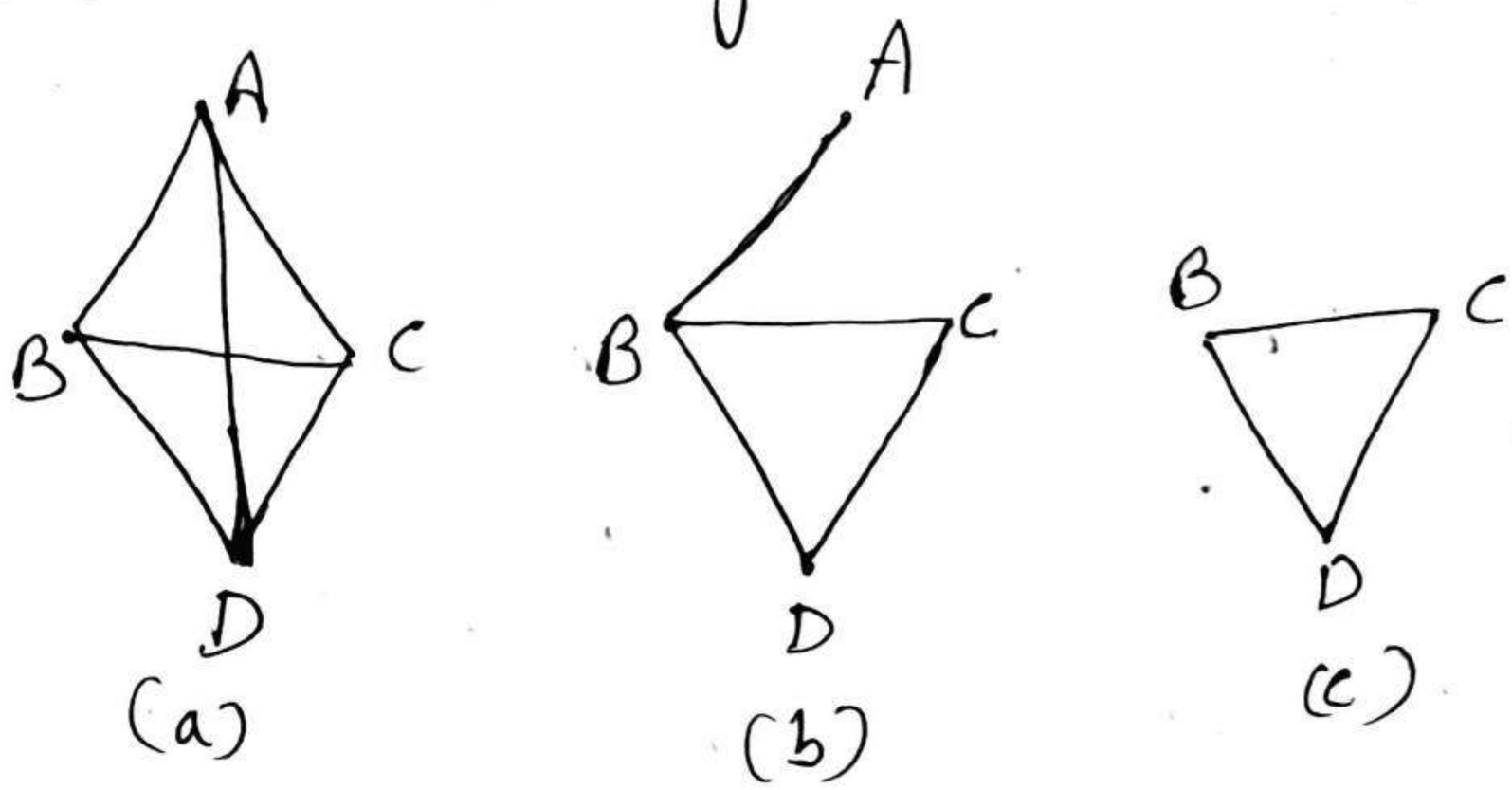
5) A single edge in a graph G, together with its end vertices, is a subgraph of G.

Scanned by TapScanner

Spanning subgraph:

Given a graph $G=(V,E)$, if there is a subgraph $G_1=(V_1,E_1)$ of G such that $V_1=V$, then G_1 is called a spanning subgraph of G.

Ex:

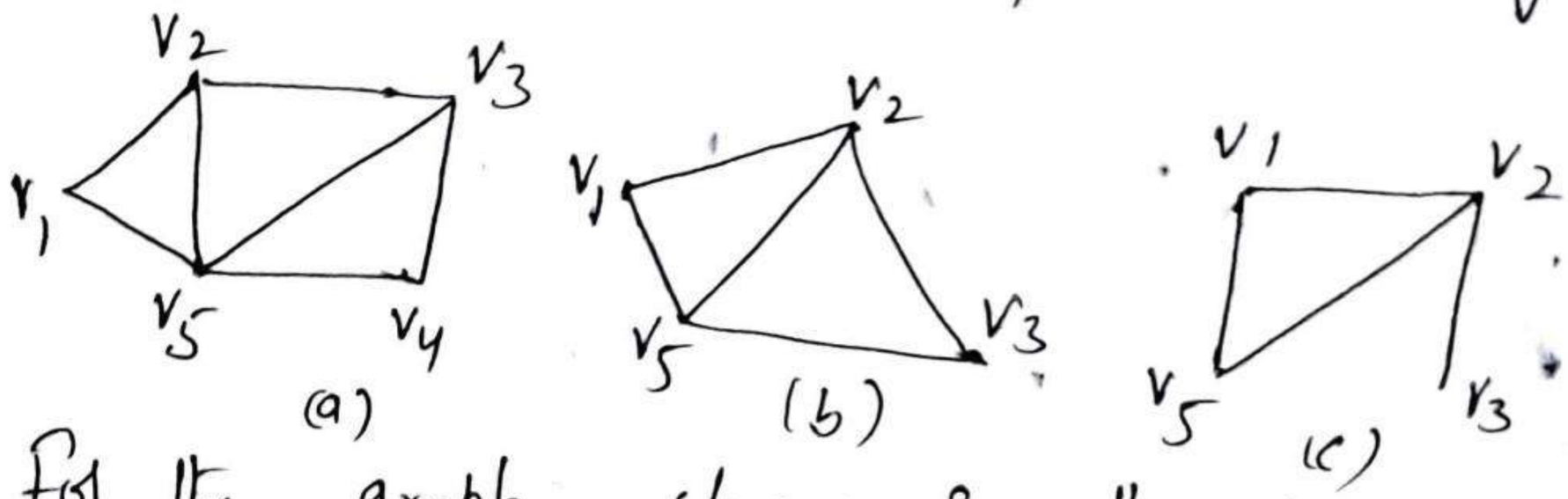


In other words, a subgraph G_1 of a graph G is a spanning subgraph of G whenever the vertex set of G_1 contains all vertices of G. Thus, a graph and all its spanning subgraphs have the same vertex set. Every graph is its own spanning subgraph.

For example, for the graph shown in fig (a), the graph shown in fig (b) is a spanning subgraph whereas the graph shown in (c) is a subgraph but not a spanning subgraph.

Scanned by TapScanner

Induced subgraph: Given a graph $G = (V, E)$; suppose there is a subgraph $G_1 = (V_1, E_1)$ of G such that every edge $\{A, B\}$ of G, where $A, B \in V_1$ is an edge of G_1 also. Then G_1 is called a subgraph of G induced by V_1 and is denoted by $\langle V_1 \rangle$



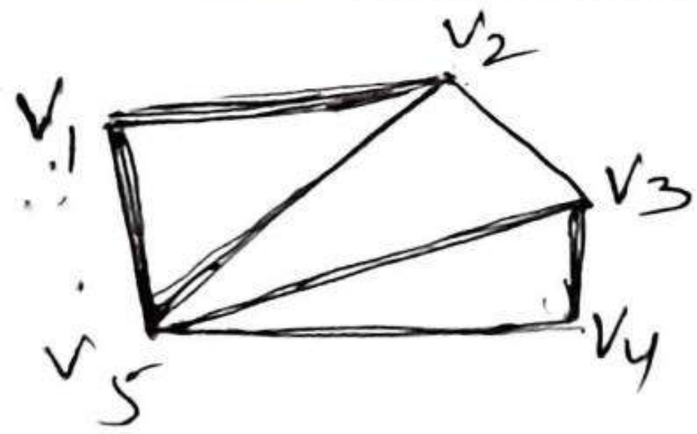
For the graph shown in the fig(a), the graph shown in fig(b) is an induced subgraph — induced by the set of vertices $V_1 = \{v_1, v_2, v_3, v_5\}$ whereas the graph shown in fig(c) is not an induced subgraph.

Edge-disjoint and vertex-disjoint subgraphs

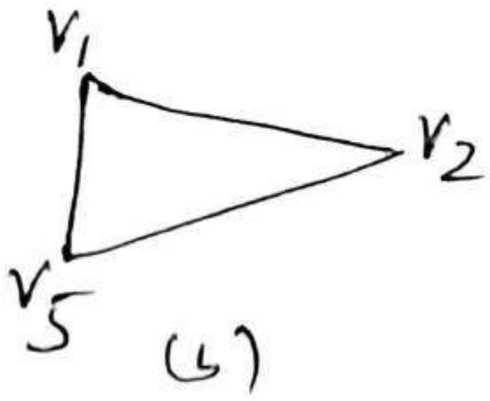
Let G be a graph and G_1 and G_2 be two subgraphs of G. Then

(1) G_1 and G_2 are said to be edge-disjoint if they do not have any common edge.

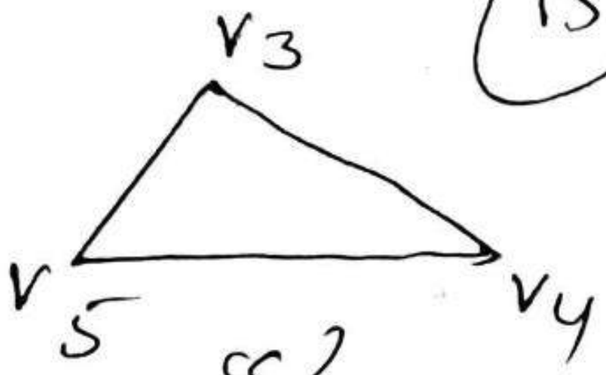
(2) G_1 and G_2 are said to be vertex-disjoint if they do not have any common edge and any common vertex.



(a)



(b)



(c)

fig (a), (b), (c) are edge-disjoint but not vertex-disjoint subgraphs.

Operations on Graphs:

UNION OF TWO ~~subgraph~~ graphs:

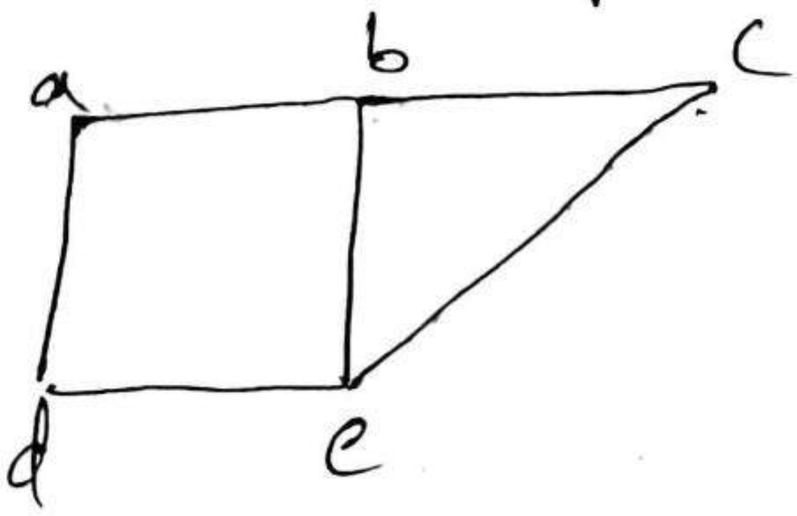
Consider two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$. Then the graph whose vertex set is $V_1 \cup V_2$ and the edge set is $E_1 \cup E_2$ is called the union of G_1 and G_2; it is denoted by $G_1 \cup G_2$.

Thus,

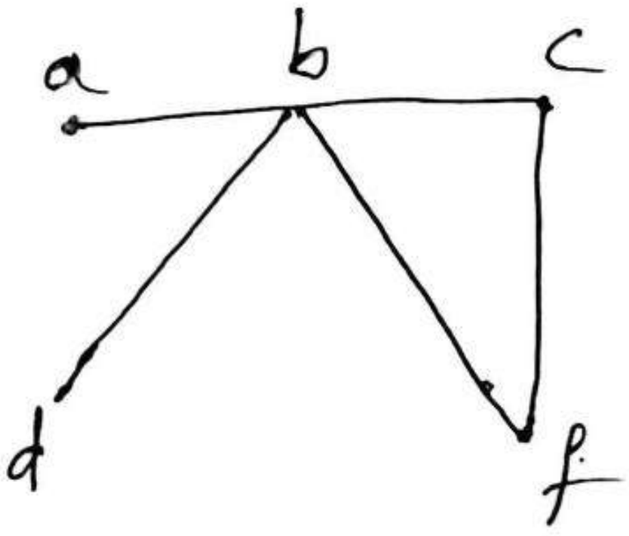
$$G_1 \cup G_2 = (V_1 \cup V_2, E_1 \cup E_2).$$

Ex:

1) Find the union of the two graphs



G_1



G_2

Scanned by TapScanner

Sol: Here $V_1 = \{a, b, c, d, e\}$ and $V_2 = \{a, b, c, d, f\}$

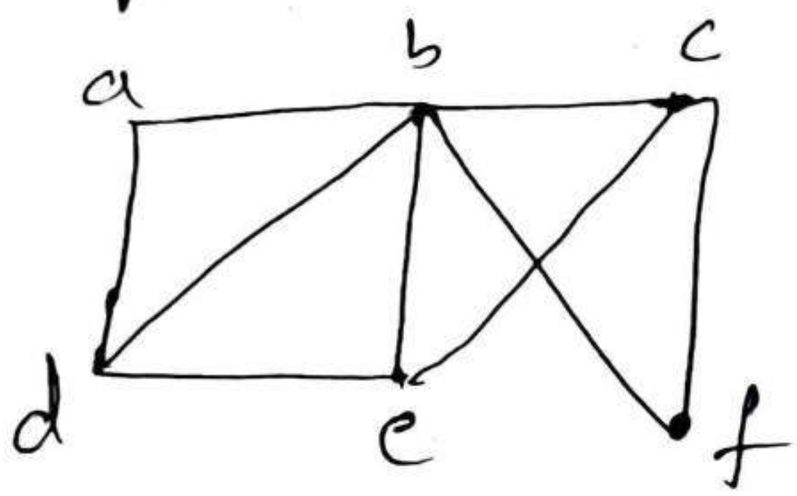
$\therefore V_1 \cup V_2 = \{a, b, c, d, e, f\}$

Similarly, $E_1 = \{(a,b), (b,c), (a,d) (d,e) (b,e) (c,e)\}$

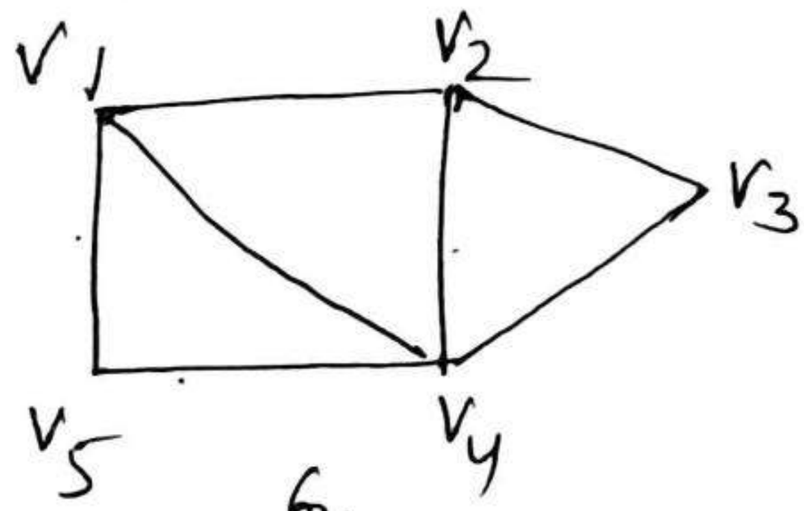
$E_2 = \{(a,b), (b,c), (c,f), (b,d), (b,f)\}$

$\therefore E_1 \cup E_2 = \{(a,b), (b,c) (a,d), (d,e), (b,e), (c,e)$
$(c,f), (b,d), (b,f)\}$

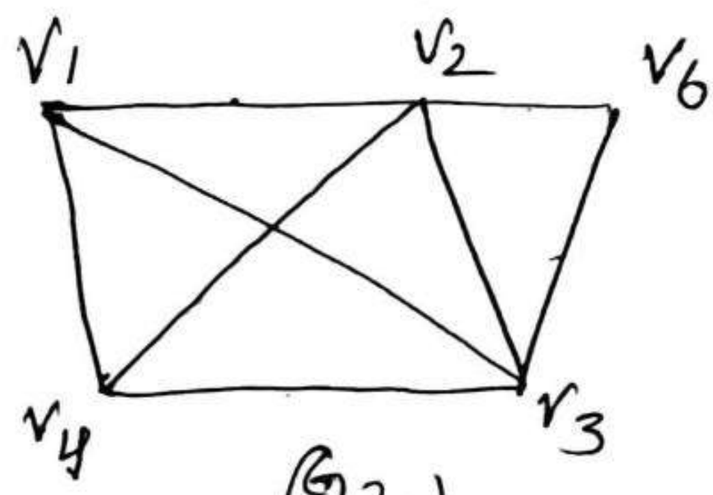
$G_1 \cup G_2$ is given by



2)



G_1



(G_2)

Sol:

Here $V(G_1) = \{v_1, v_2, v_3, v_4, v_5\}$ $V(G_2) = \{v_1, v_2, v_3, v_4, v_6\}$

$\therefore V(G_1) \cup V(G_2) = \{v_1, v_2, v_3, v_5, v_5, v_6\}$

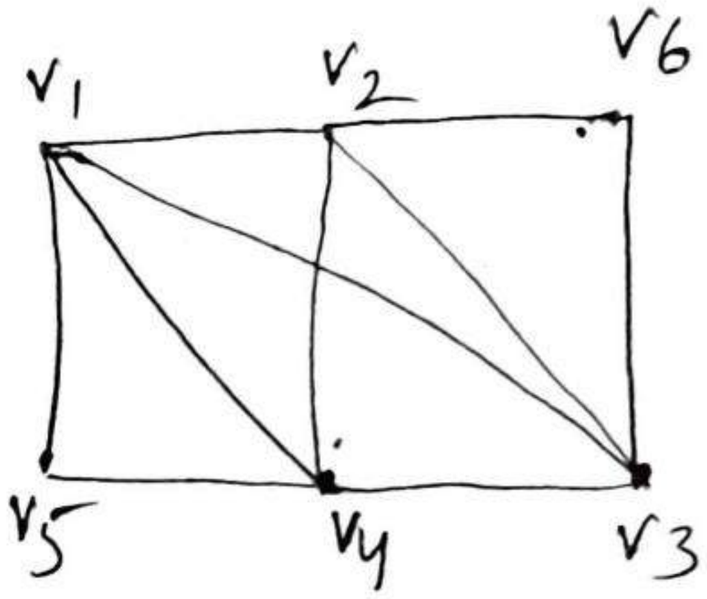
Ily $E_1 = \{(v_1, v_2) (v_2, v_3) (v_1, v_5) (v_2, v_4) (v_3, v_4) (v_1, v_4) (v_4, v_5)\}$

$E_2 = \{(v_1, v_2) (v_1, v_4), (v_1, v_3) (v_2, v_3) (v_2, v_4), (v_2, v_6)$
$(v_3, v_4) (v_3, v_6)\}$

$\therefore E_1 \cup E_2 = \{(v_1, v_2) (v_2, v_3) (v_1, v_5) (v_2, v_4) (v_3, v_4) (v_4, v_5)$
$(v_1, v_4) (v_1, v_3), (v_2, v_6) (v_3, v_6)\}$.

Scanned by TapScanner

$G_1 \cup G_2$ is



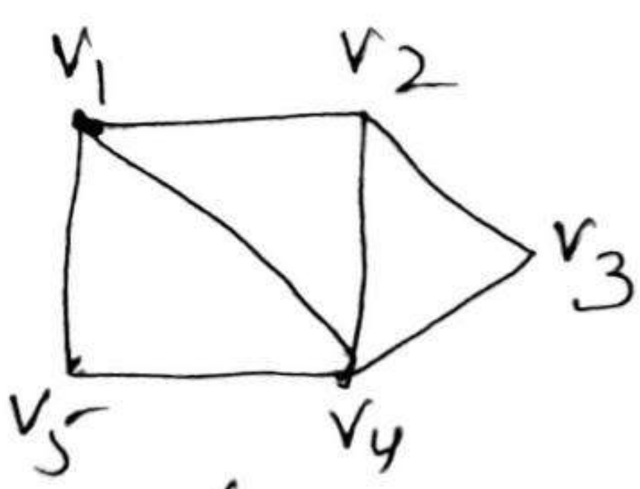
$G_1 \cup G_2$.

Intersection of Graphs

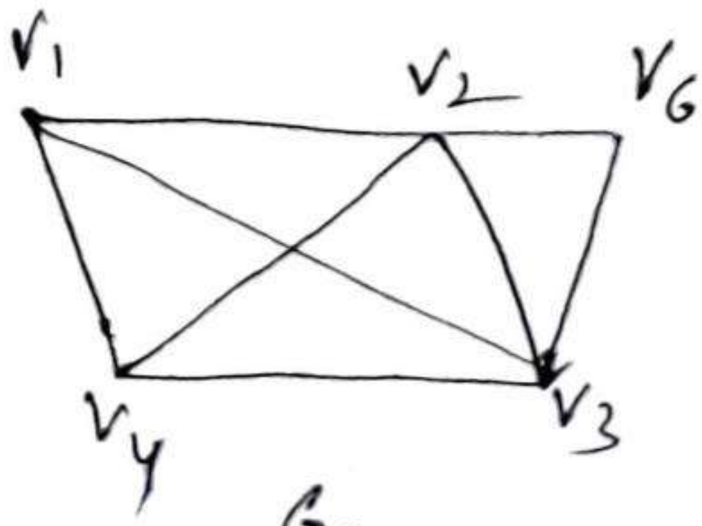
Let $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ be two graphs. Then the graph whose vertex set is $V_1 \cap V_2$ and the edge set is $E_1 \cap E_2$ is called the intersection of G_1 and G_2; it is denoted by $G_1 \cap G_2$. Thus

$$G_1 \cap G_2 = (V_1 \cap V_2, E_1 \cap E_2) \text{ if } V_1 \cap V_2 \neq \emptyset.$$

Ex: consider the two graphs



G_1



G_2.

Sol. Here $V(G_1) = \{v_1, v_2, v_3, v_4, v_5\}$ & $V(G_2) = \{v_1, v_2, v_3, v_4, v_6\}$

$V(G_1) \cap V(G_2) = \{v_1, v_2, v_3, v_4\}$

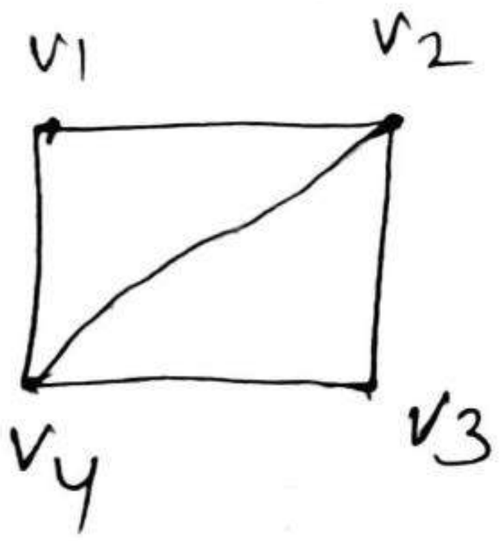
Ily

$E_1 = \{(v_1,v_2)\ (v_2,v_3)\ (v_1,v_5)\ (v_2,v_4)\ (v_3,v_4)\ (v_1,v_4)\ (v_4,v_5)\}$

$E_2 = \{(v_1,v_2)\ (v_1,v_4)\ (v_1,v_3)\ (v_2,v_3)\ (v_2,v_4)\ (v_2,v_6)\ (v_3,v_4)\ ,\ (v_3,v_6)\}$

$\therefore E_1 \cap E_2 = \{(v_1,v_2), (v_2,v_3)\ (v_2,v_4)\ (v_3,v_4)\ (v_1,v_4)\}$

$G_1 \cap G_2$ is



<u>Decomposition:</u> we say that a graph G is decomposed (or partitioned) into two subgraphs G_1 and G_2 if $G_1 \cup G_2 = G$ & $G_1 \cap G_2 =$ Null graph

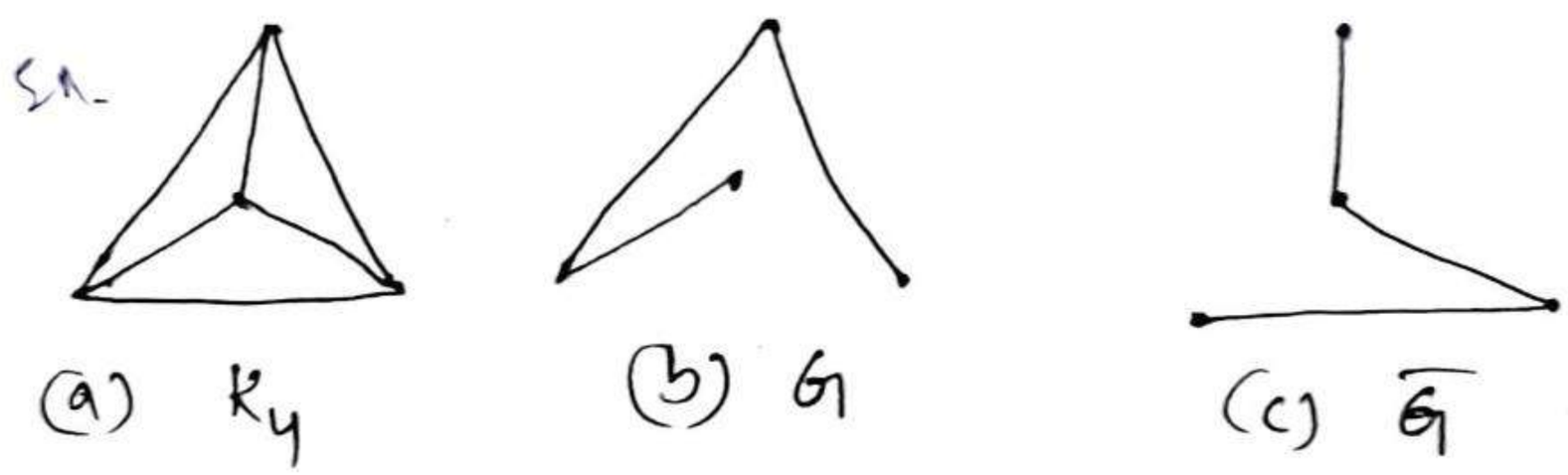
<u>COMPLEMENT OF a Graph</u>

→ we note that every simple graph of order 'n' is a subgraph of the complete graph K_n.

→ If G is a simple graph of order n, then the complement of G in K_n is called the complement of G; it is denoted by $\bar{G}$. i.e $\bar{G} = K_n - G$

Thus, the complement $\overline{G}$ of a simple graph G with 'n' vertices is that graph which is obtained by deleting those edges of K_n which belong to G. Thus $\overline{G} = K_n - G$.

Ex: 1) In figure (a), the complete graph K_4 is shown. A simple graph G of order 4 is shown in fig (b). The complement, $\overline{G}$ of G is shown in fig (c).

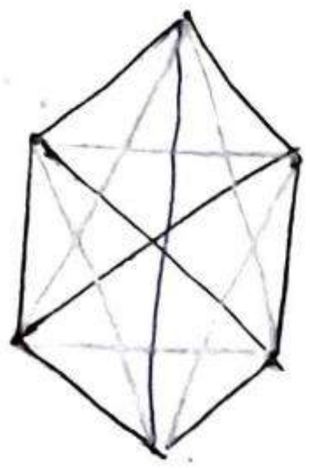


(a) K_4 (b) G (c) $\overline{G}$.

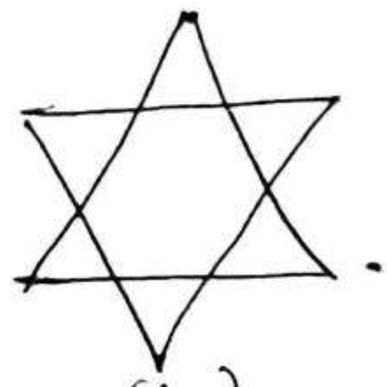
$\overline{G} = K_4 - G$

Observe that $G, \overline{G}$ and K_4 have the same vertices and that the edges of $\overline{G}$ are got by deleting those edges from K_4 which belong to G.

2) In figure (a), a graph G of order 6 is shown as a subgraph of K_6, the edges of G being shown in thick lines, Its complement $\overline{G}$ is shown in fig (b). The graph shown in fig (b) is known as the David graph.



(a)



(b)

K_6

G = thick lines

$\bar{G} = K_6 - G$

Walks and their classification:

In this section, we consider five important subgraphs of a graph, called a walk, a trail, a circuit, a path and a cycle.

Walk:

consider a graph G having at least one edge. In G, consider a finite, alternating sequence of vertices and edges of the form

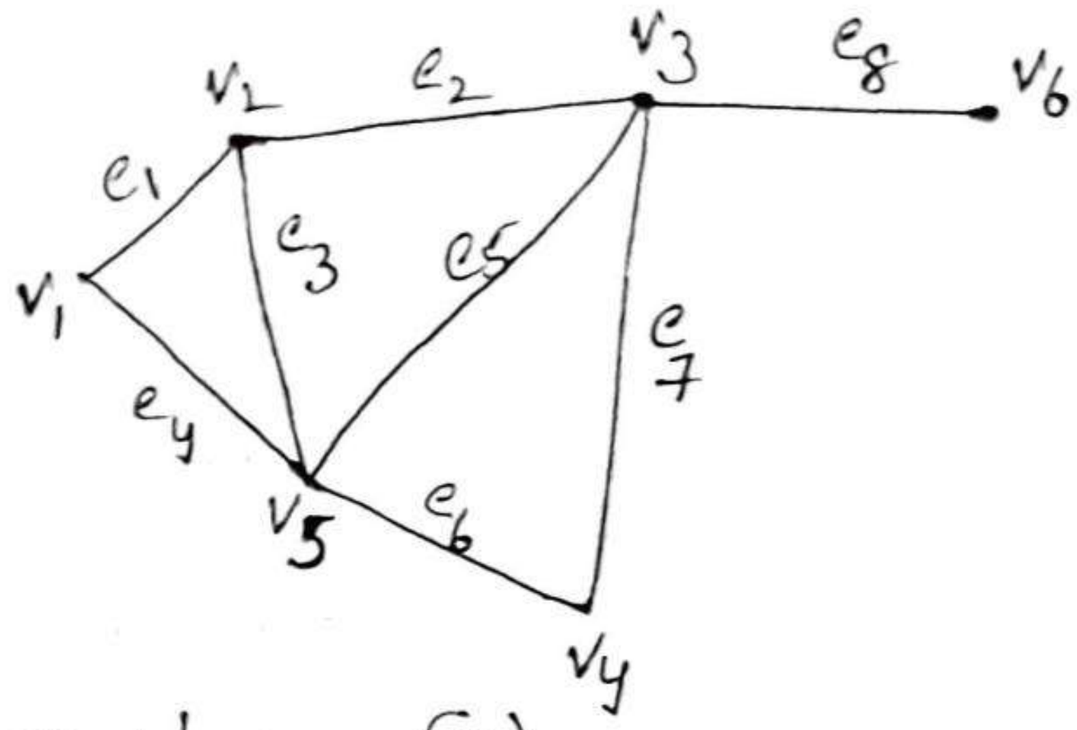
$$v_i e_j v_{i+1} e_{j+1} v_{i+2} \ldots e_k v_m .$$

which begins and ends with vertices and which is such that each edge in the sequence is incident on the vertices preceding and following it in the sequence. Such a sequence is called a walk in G.

→ In a walk, a vertex or an edge (or both) can appear more than once.

→ The no. of edges present in a walk is called its length.

Ex: consider the graph shown below:



In this graph: (a)

(i) The sequence $v_1 e_1 v_2 e_2 v_3 e_8 v_6$ is a walk of length 3. (because, this walk contains 3 edges: e_1, e_2, e_8). In this walk, no vertex and no edge is repeated.

(ii) The sequence $v_1 e_4 v_5 e_3 v_2 e_2 v_3 e_5 v_5 e_6 v_4$ is a walk of length 5. In this walk, the vertex v_5 is repeated; but no edge is repeated.

(iii) The sequence $v_1 e_1 v_2 e_3 v_5 e_3 v_2 e_2 v_3$ is a walk of length 4. In this walk, the edge e_3 is repeated and the vertex v_2 is repeated.

→ The vertex with which a walk begins is called the <u>initial vertex</u> (or the origin) of the walk and the vertex with which a walk ends is called the <u>final vertex</u> (or the terminus) of the walk. The initial vertex and the final vertex of a walk are together

called its terminal vertices. The terminal vertices of a walk need not be distinct.

→ A walk that begins and ends at the same vertex is called a closed walk.

In other words, a closed walk is a walk in which the terminal vertices are coincident.

In the graph(a) shown above, $v_1 e_1 v_2 e_3 v_5 e_4 v_1$ is a closed walk.

→ A walk that is not closed is called an open walk. In other words, an open walk is a walk that begins and ends at two different vertices.

In the graph, $v_1 e_1 v_2 e_2 v_3 e_5 v_5$ is an open walk.

Trail and circuit:

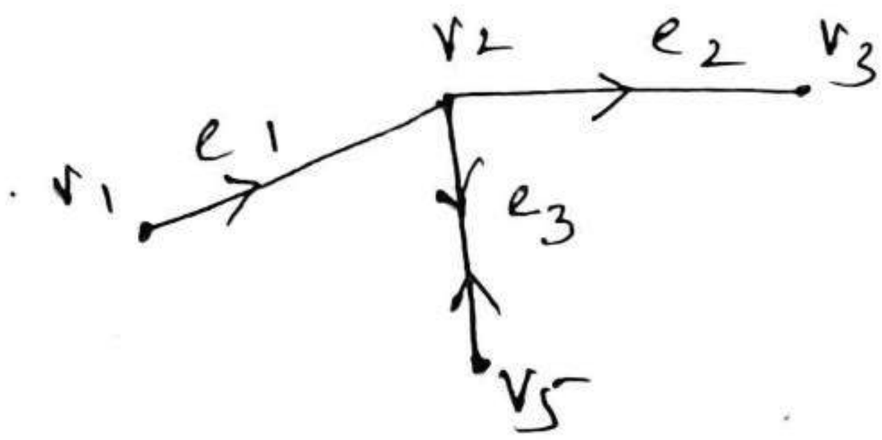
In a walk, vertices and/or edges may appear more than once.

→ If in an open walk no edge appears more than once, then the walk is called a trail.

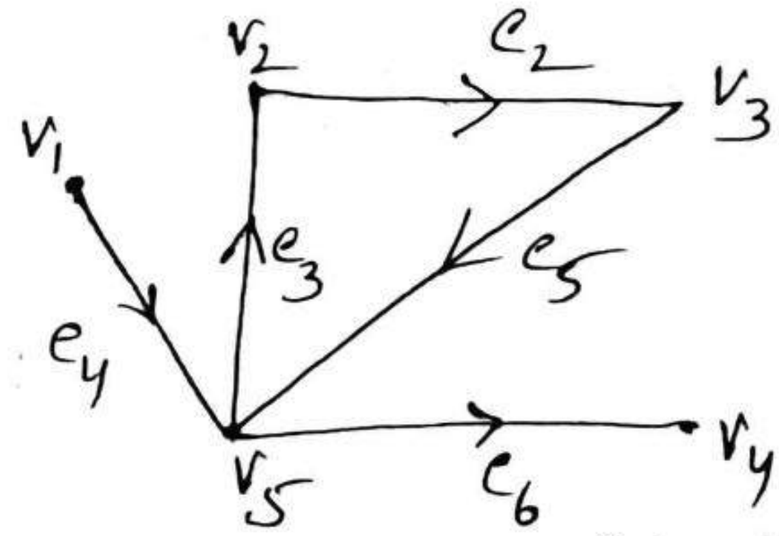
→ A closed walk in which no edge appears more than once is called a circuit.

Scanned by TapScanner

→ for example, in the above figure (a), the open walk $v_1 e_1 v_2 e_3 v_5 e_3 v_2 e_2 v_3$ is not a trail (because in this walk, the edge e_3 is repeated) where as the open walk $v_1 e_4 v_5 e_3 v_2 e_2 v_3 e_5 v_5 e_6 v_4$ is a trail. (in fig (b))

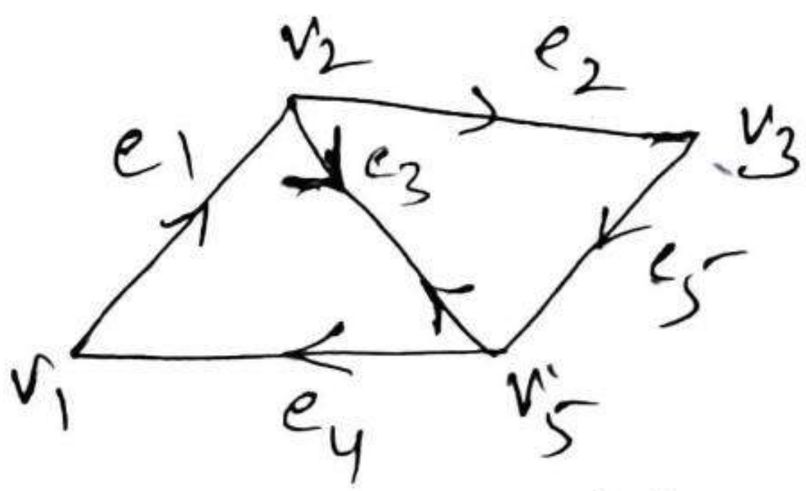


(a) Not a trail



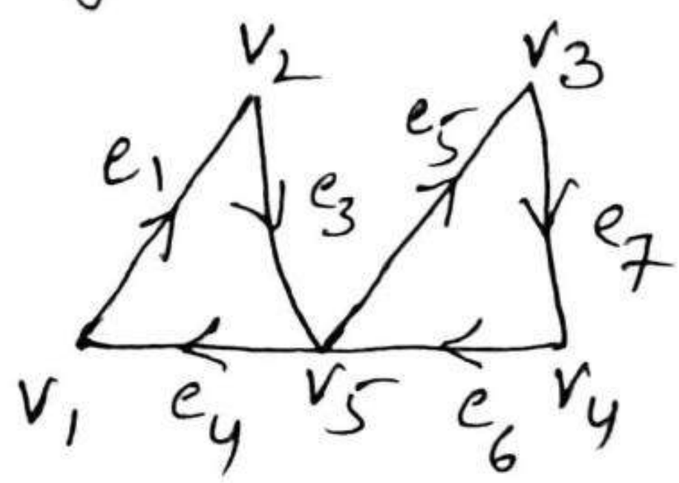
(b) Trail.

Also, in the same figure (c), the closed walk $v_1 e_1 v_2 e_3 v_5 e_3 v_2 e_2 v_3 e_5 v_5 e_4 v_1$ is not a circuit (because e_3 is repeated) where as the closed walk $v_1 e_1 v_2 e_3 v_5 e_5 v_3 e_7 v_4 e_6 v_5 e_4 v_1$ is a circuit. (fig d).



Not a ckt.

fig (c)



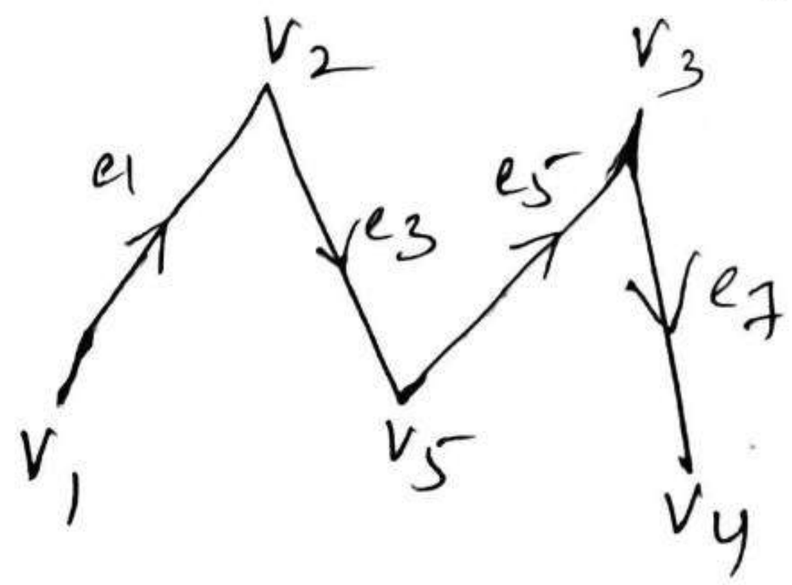
circuit.

fig (d)

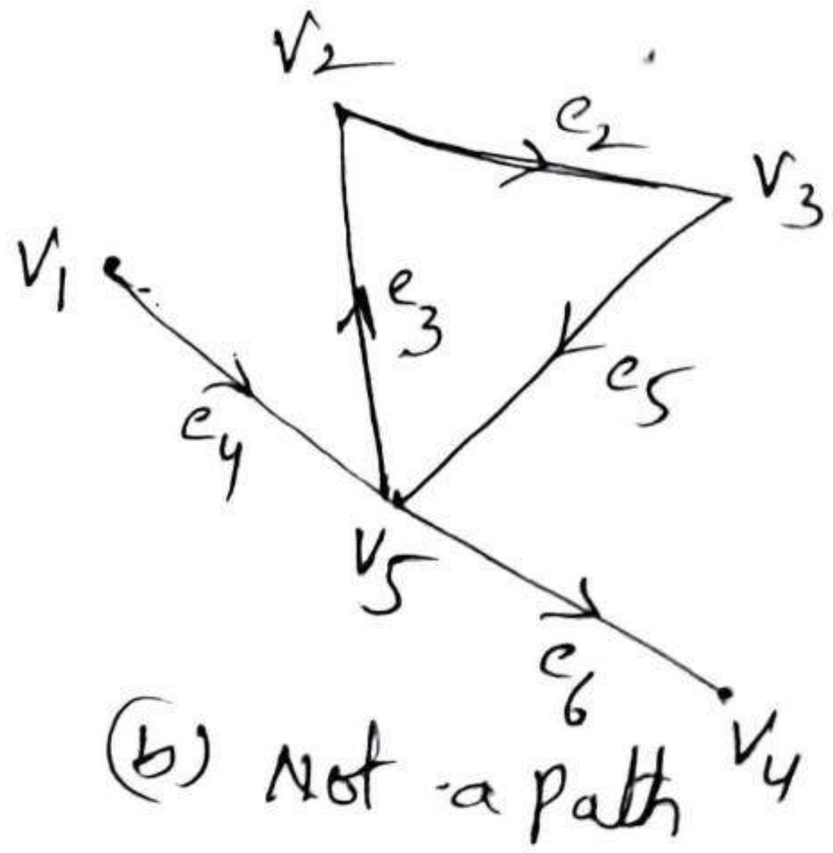
Path and cycle

A trail in which no vertex appears more than once is called <u>a path</u>.

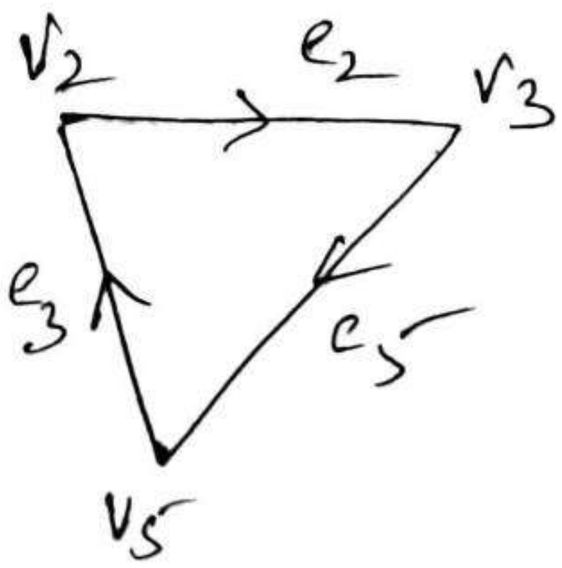
A circuit in which the terminal vertex does not appear as an internal vertex (also) and no internal vertex is repeated is called a cycle.



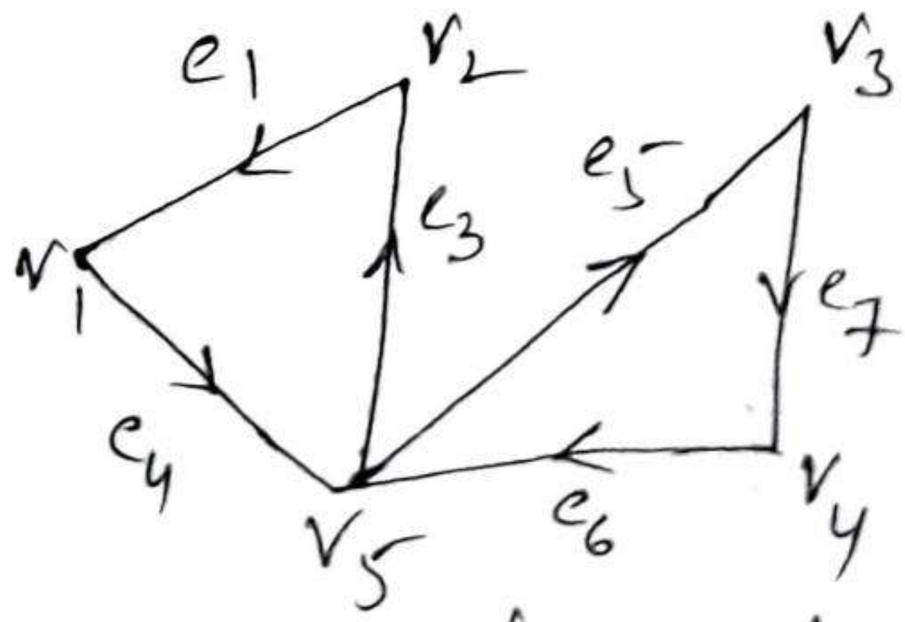
(a) Path



(b) Not a path



(c) cycle



(d) Not a cycle.

For example, in the above fig, the trail $v_1 e_1 v_2 e_3 v_5 e_5 v_3 e_7 v_4$ is a path whereas the trail $v_1 e_4 v_5 e_3 v_2 e_2 v_3 e_5 v_5 e_6 v_4$ is not a path (because in this trail,

v_5 appears twice).

Also in the same figure, the circuit $v_2 e_2 v_3 e_5 v_5 e_3 v_2$ is a cycle whereas the circuit $v_2 e_1 v_1 e_4 v_5 - e_5 - v_3 e_7 v_4 e_6 v_5 e_3 v_2$ is not a cycle (because, in this circuit, v_5 appears twice).

The following facts are to be emphasized.

1) A walk can be open or closed. In a walk (closed or open), a vertex and/or an edge can appear more than once.

2) A trail is an open walk in which a vertex can appear more than once but an edge cannot appear more than once.

3) A circuit is a closed walk in which a vertex can appear more than once but an edge cannot appear more than once.

4.) A path is an open walk in which neither a vertex nor an edge can appear more than once. Every path is a trail; but a trail need not be a path.

5) A cycle is a closed walk in which neither a vertex nor an edge can appear more than once. Every cycle is a circuit, but a circuit need not be a cycle.

6) If a cycle contains only one edge, it has to be a loop.

7) Two parallel edges (when they occur) form a cycle. (closed)

8) In a simple graph, a cycle must have at least three edges. (A cycle formed by three edges is called a triangle).

Scanned by TapScanner

Connected and Disconnected graph

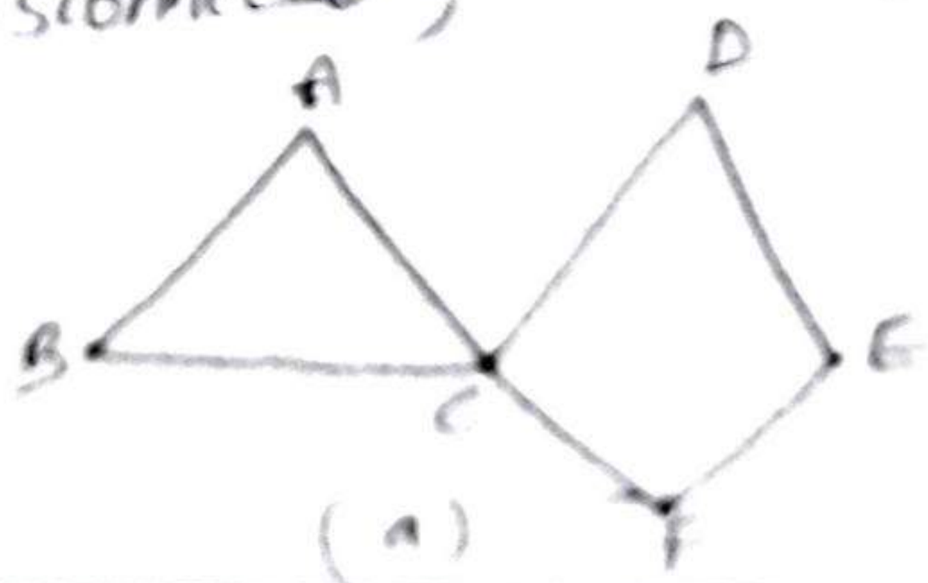
Q1

Consider a graph G of order greater than or equal to two. Two vertices in G are said to be connected if there is at least one path from one vertex to the other.

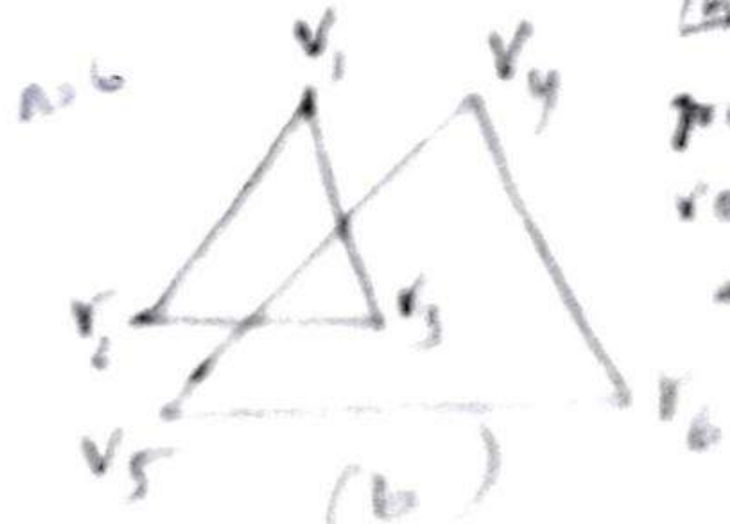
We say that G is connected if every pair of distinct vertices in G are connected. otherwise, G is called a disconnected graph.

In other words, a graph G is said to be
(i) connected if there is at least one path between every two distinct vertices in G, and
(ii) disconnected if G has at least one pair of distinct vertices between which there is no path.

For example, the graph shown in fig (a) is connected whereas the graph shown in fig (b) is disconnected; because, there is no path from v_1 to v_6



(a)



(b)

In A connected graph with n vertices has at least (n-1) edges

Euler circuits and Euler trails

Consider a connected graph G. If there is a circuit in G that contains all the edges of G, then that circuit is called an <u>Euler circuit</u> (or Eulerian line, or Euler tour) in G.

→ If there is a trail in G that contains all the edges of G, then that trail is called an <u>Euler trail</u> (or unicursal line) in G.

→ We know that in a trail and a ckt no edge can appear more than once but a vertex can appear more than once. This property is carried to Euler trails and Euler circuits also.

→ Since Euler circuit and Euler trails include all the edges, they automatically should include all vertices as well.

→ A connected graph that contains an Euler circuit is called an Euler graph or Eulerian graph. A connected graph that contains an Euler trail is called a semi-Euler graph (or a semi-Eulerian graph or unicursal graph).

Scanned by TapScanner

For Example, in the graph shown in fig (a), the closed walk

$Pe_1Qe_2Re_3Pe_4Se_5Re_6Te_7P$ is an Euler circuit.

Therefore, this graph is an Euler graph.

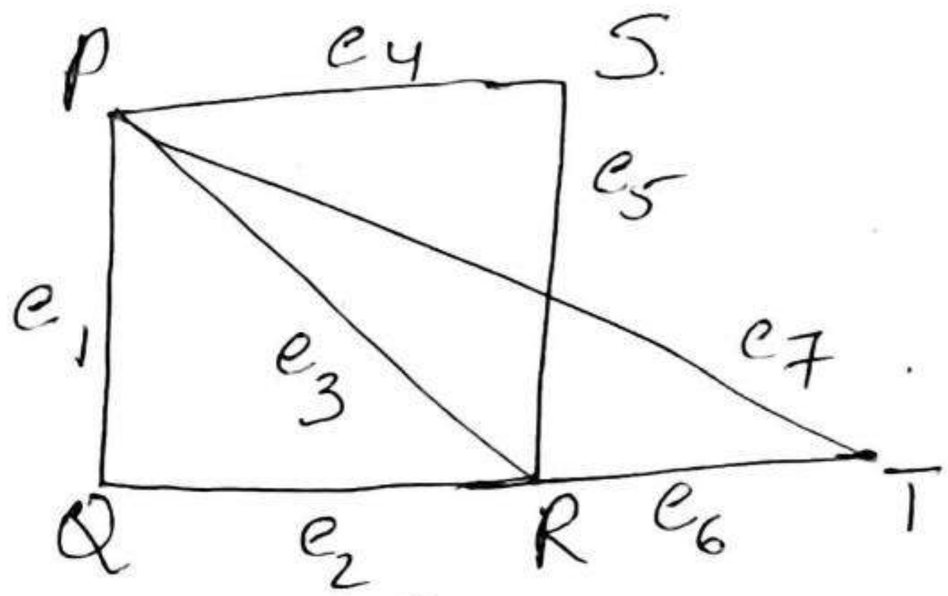


fig (a)

Consider the graph shown in fig (b). We observe that, in this graph, every sequence of edges which starts and ends with the same vertex and which includes all edges will contain at least one repeated edge. Thus the graph has no Euler circuit. Hence this graph is not an Euler graph.

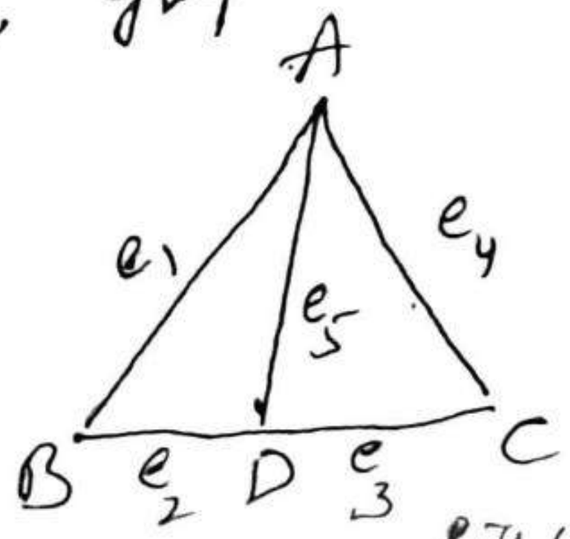


fig (b)

It may be seen that the trail $Ae_1 Be_2 De_3 Ce_4 Ae_5 D$
$Ae_1 Be_2 De_3 Ce_4 Ae_5 D$
in the graph is an Euler trail. This graph is therefore a semi-Euler graph.

537 JLMott

Note: An Euler Path is a multigraph is a Path that includes each edge of the multigraph exactly once and intersects each vertex of the multigraph at least once.

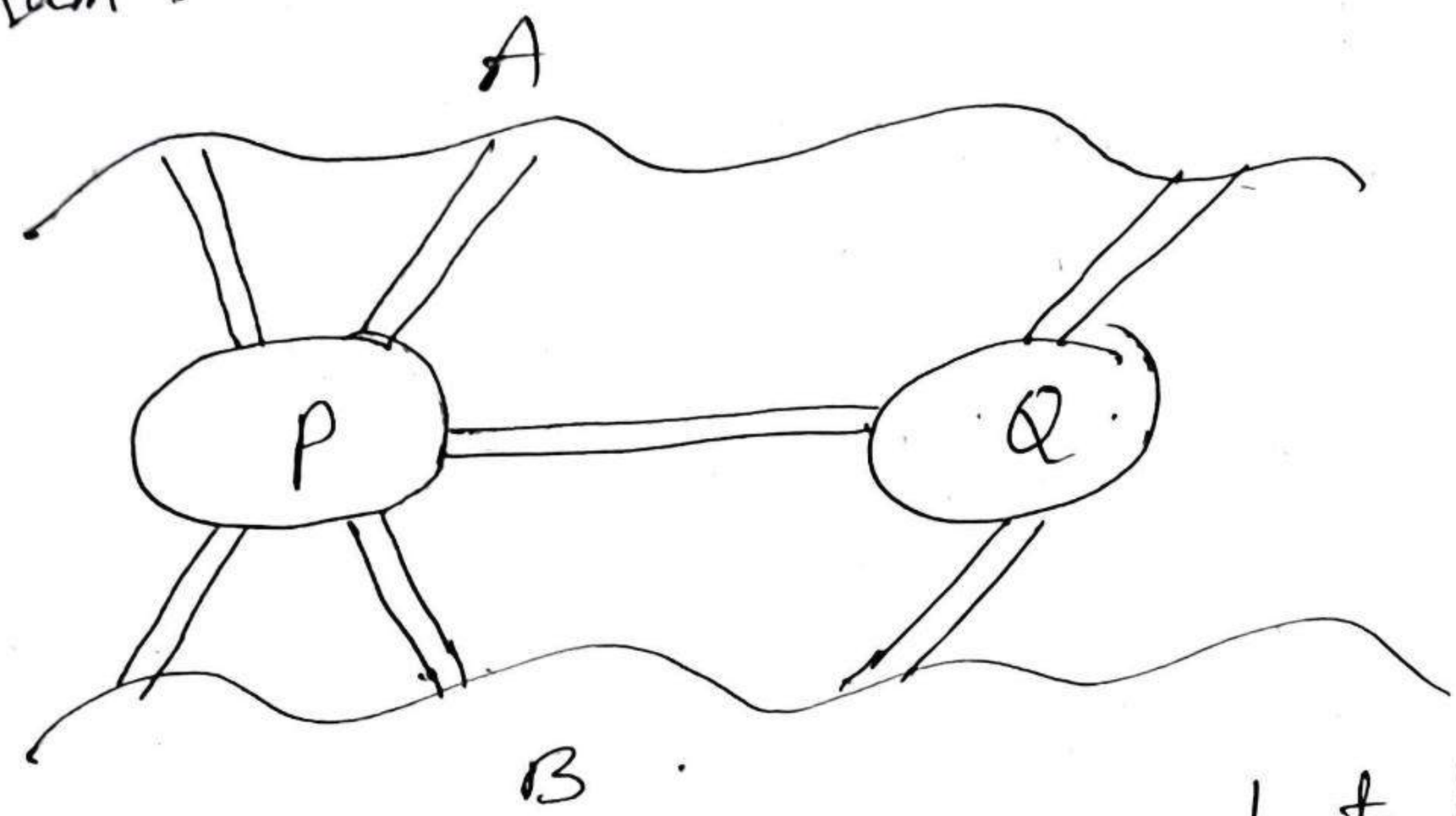
Thm 1: A connected graph G has an Euler circuit (i.e G is an Euler graph) if and only if all vertices of G are of even degree.

Th2: A connected graph G has an Euler circuit (i.e, G is an Euler graph) if and only if G can be decomposed into edge-disjoint cycles.

→ The Konigsberg Bridge Problem:

In the eighteenth century city named Konigsberg in East Prussia (Europe), there flowed a river named Pregel river which divided the city into four Party. Two of these Party were the banks of the river and two were islands. These party were connected with each other through seven bridges as shown in figure. i.e, There were two islands 'P' and 'Q' connected

to each other and to the banks 'A' and 'B' with seven bridges as shown in the following diagram.



The citizens of the city seemed to have posed the following problem:

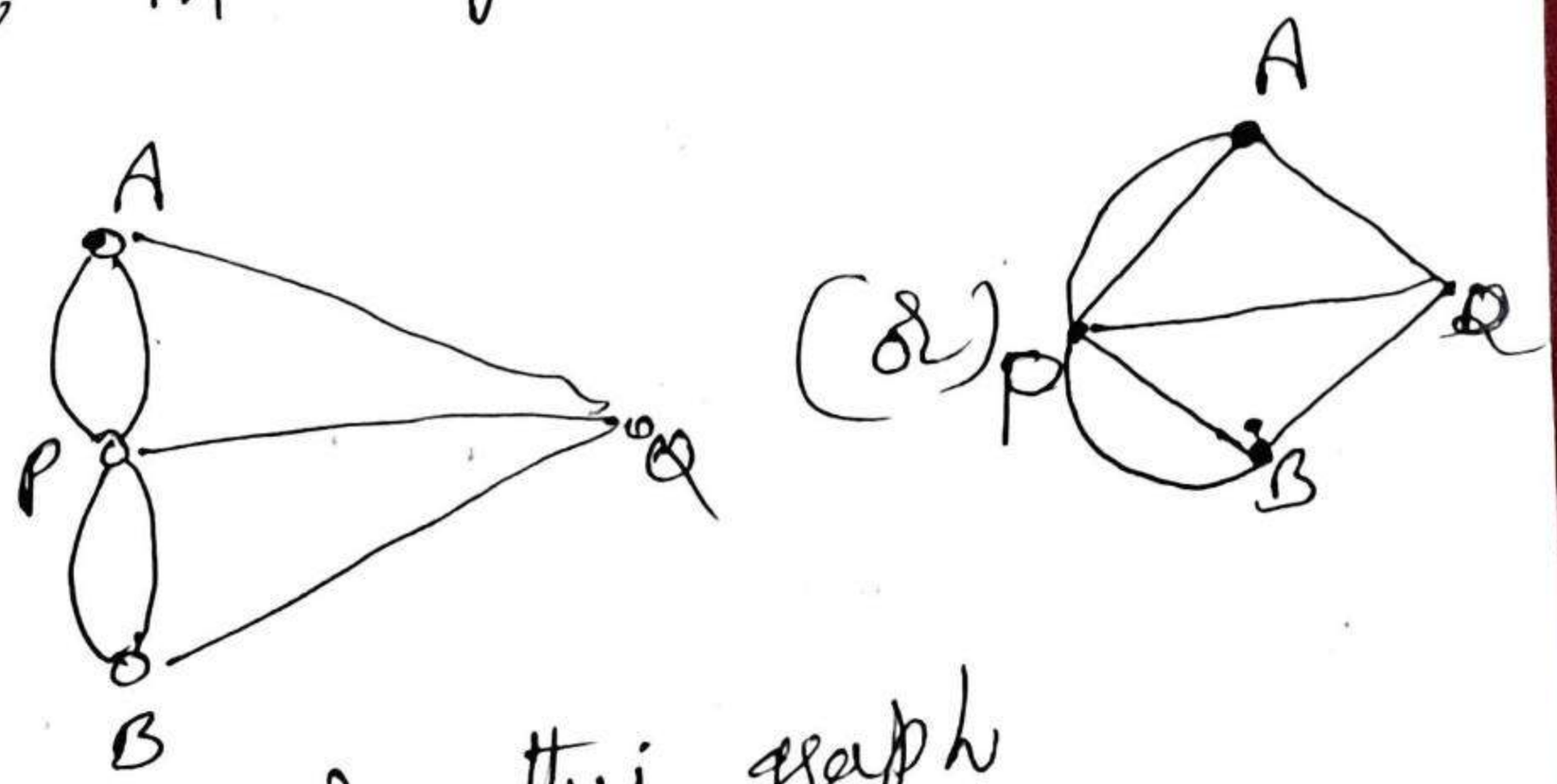
By starting at any of the four land areas can we return to that area after crossing each of the seven bridges exactly once?

This problem, now known as the Konigsberg Bridge problem, remained unsolved for several years.

In the year 1736, Euler analysed the problem with the help of a graph and gave the solution. This was indeed the starting point for the

development of graph theory.

Let us see what the solution is (as given by Euler). Denote the land areas of the city by A, B, P, Q, where A, B are the banks of the river and P, Q are the islands (see the fig) construct a graph by treating the four land areas as four vertices and the seven bridges connecting them as seven edges. The graph is shown in fig



We note that, in this graph

$$\deg(A) = \deg(B) = \deg(Q) = 3,$$

$$\deg(P) = 5.$$

which are not even. Therefore the graph does not have an Euler circuit. This means that there does not exist a closed

walk that contains all the edges exactly (24)
walk : This amounts to saying that it is
once
not possible to walk over each of the seven
bridges exactly once and return to the starting
point.

Euler proved that a solution for this problem
does not exist.

Hamilton cycles and Hamilton paths:

Let G be a connected graph. If there is a
cycle in G that contains all the vertices of G,
then that cycle is called a Hamilton cycle in
G.

→ A Hamilton cycle (when it exists) in a graph
of n vertices consists of exactly 'n' edges.

Because, a cycle with 'n' vertices has 'n' edges.

→ By definition, a Hamilton cycle (when it exists)
in a graph G must include all vertices in G.
This does not mean that it should include all
edges of G.

Scanned by TapScanner

Def^n: A graph that contains a Hamilton cycle is called a Hamilton graph (or Hamiltonian graph).

(i) For example, in the graph shown in fig (1) the cycle shown in thick lines is a Hamilton cycle. (observe that this cycle does not include the edge BD). The graph is therefore a Hamilton graph.

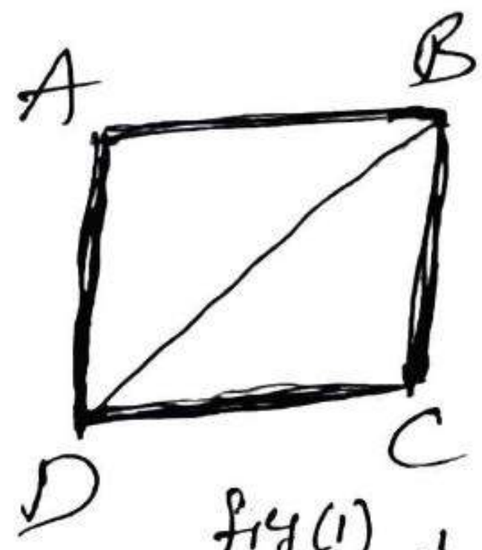
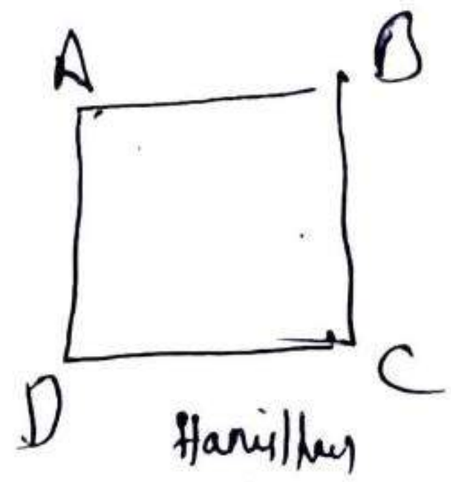


fig (1)



Hamilton

→ It is easy to see that in the hyper-cube Q_3 shown in fig (2), the cycle ABCDSRQPA is a Hamilton cycle. (observe that this cycle does not include all the edges). Therefore Q_3 is a Hamilton graph.

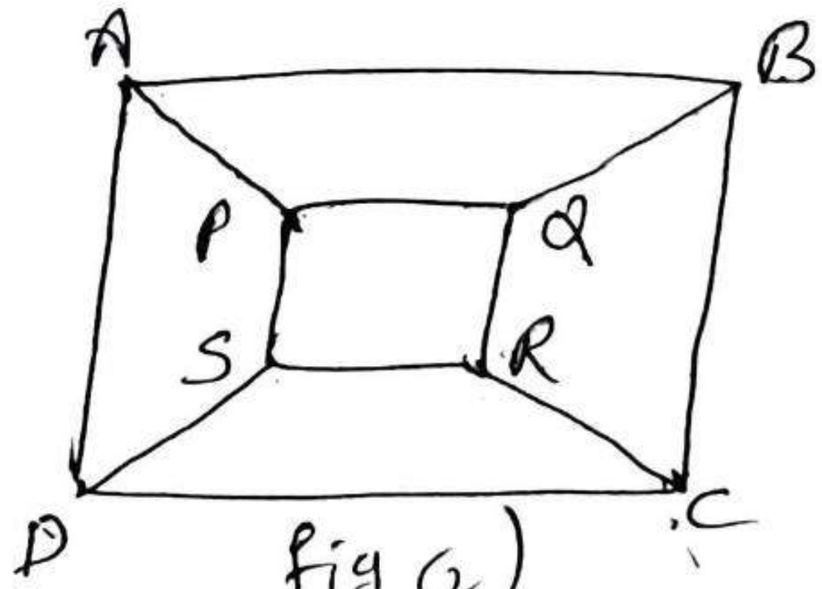
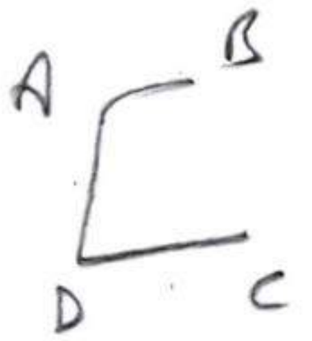


fig (2)

⇒ A path (if any) in a connected graph which includes every vertex (but not necessarily every edge) of the graph is called a <u>Hamilton/Hamiltonian path</u> in the graph.

For example, in the graph shown in fig (3), the path shown in thick lines is a Hamilton path.

BADC ✓



$n = 4$
$n-1 = 3$ edges

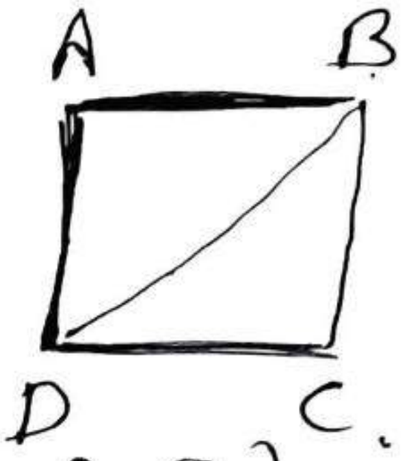
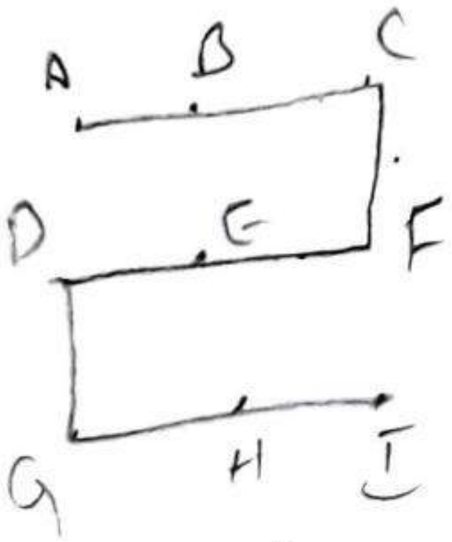
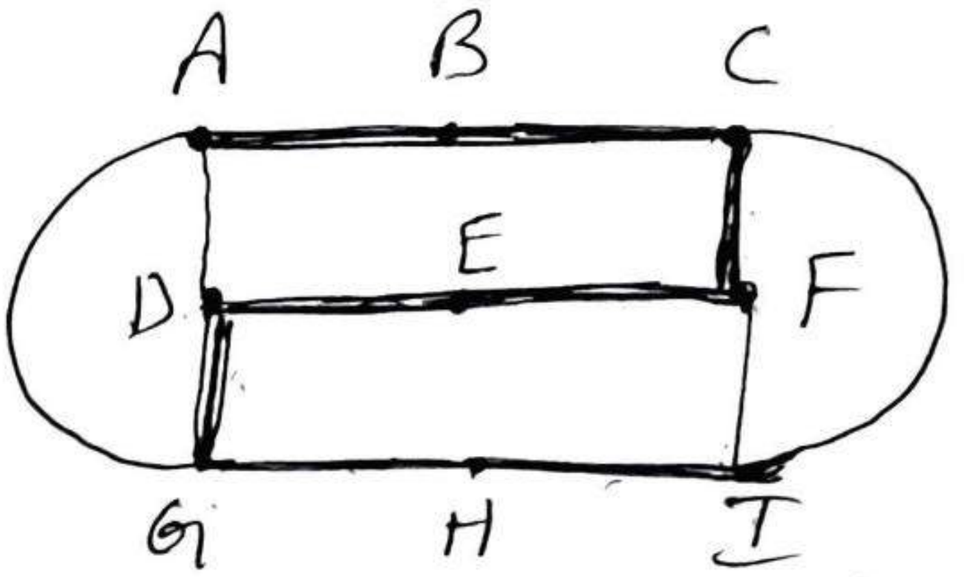


fig (3)

→ In the graph shown in fig (4), the path ABCFEDGHI is a Hamilton path. we check that this graph does not contain a Hamilton cycle.

8





$n = 9$

$n-1 = 8$ edges in Hamilton path

Since a Hamilton path in a graph G meets every vertex of G, the length of a Hamilton path (if any) in a connected graph of n vertices is (n-1).

(Recall that a path with n vertices has $(n-1)$ edges).

Scanned by TapScanner

Planar and non-planar graph

Definition: (i) A graph G is said to be a planar graph if there exists some geometric representation of G which can be drawn on a plane such that no two of its edges intersect.

NOX (ii) A graph that cannot be drawn on a plane without a cross over b/w its edges is called a non-planar graph).

(iii) NO A drawing of a geometric representation of a graph on any surface such that no edges intersect is called an embedding.

(OR)

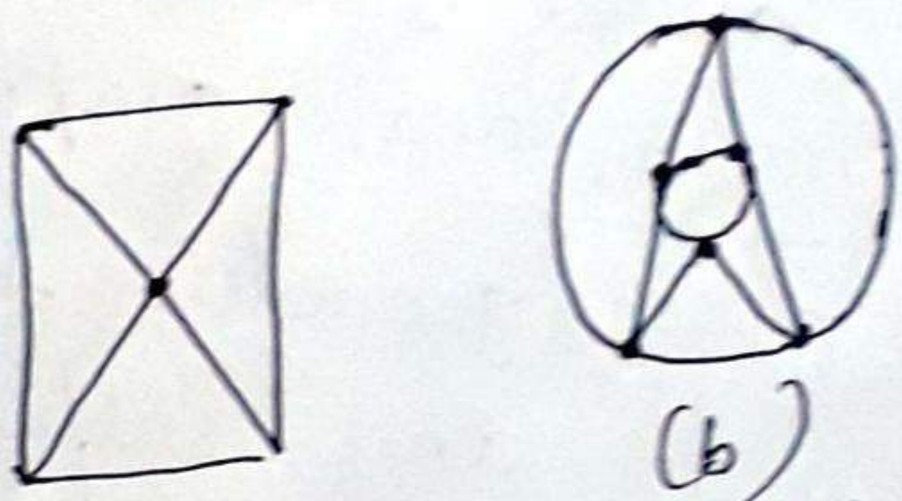
Def: A graph which can be represented by at least one plane drawing in which the edges meet only at the vertices is called a planar graph.

A graph which cannot be represented by a plane drawing in which the edges meet only

at the vertices is called a non-planar graph.

Ex: S.T (i) a graph of order 5 and size 8 (ii) a graph of order 6 and size 12 are planar.

Sol: A graph of order 5 and size 8 can be represented by a plane drawing in which the edges of the graph meet only at the vertices, as shown in fig (a). Therefore this graph is a planar graph.

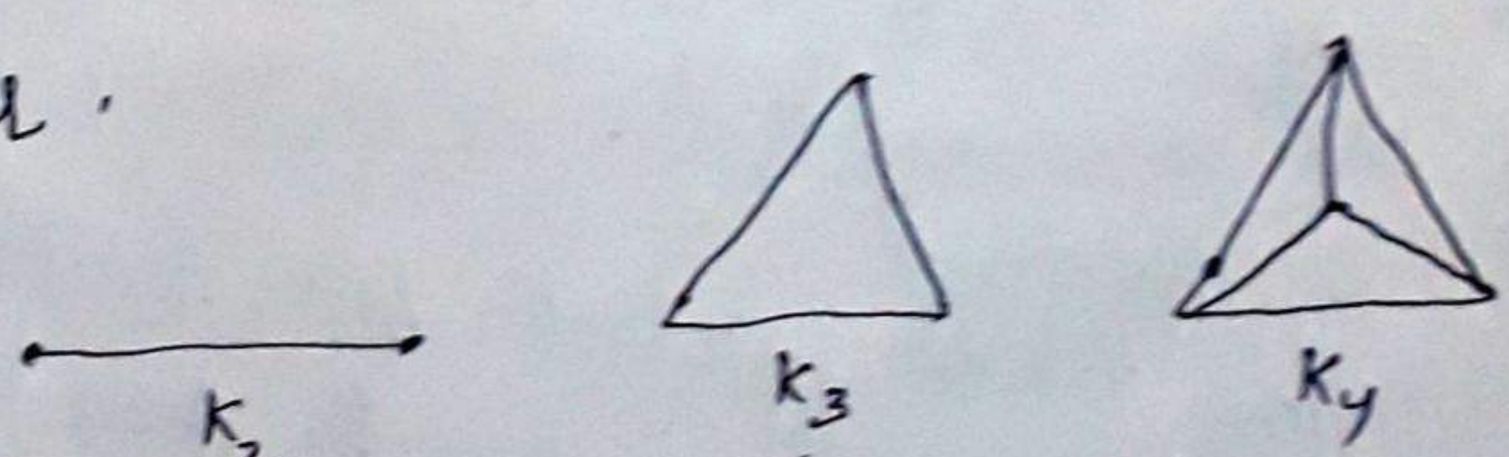


(a) (b)

Similarly, fig (b) shows that a graph of order 6 and size 12 is planar.

2) S.T the complete graphs K_2, K_3 and K_4 are planar.

Sol:



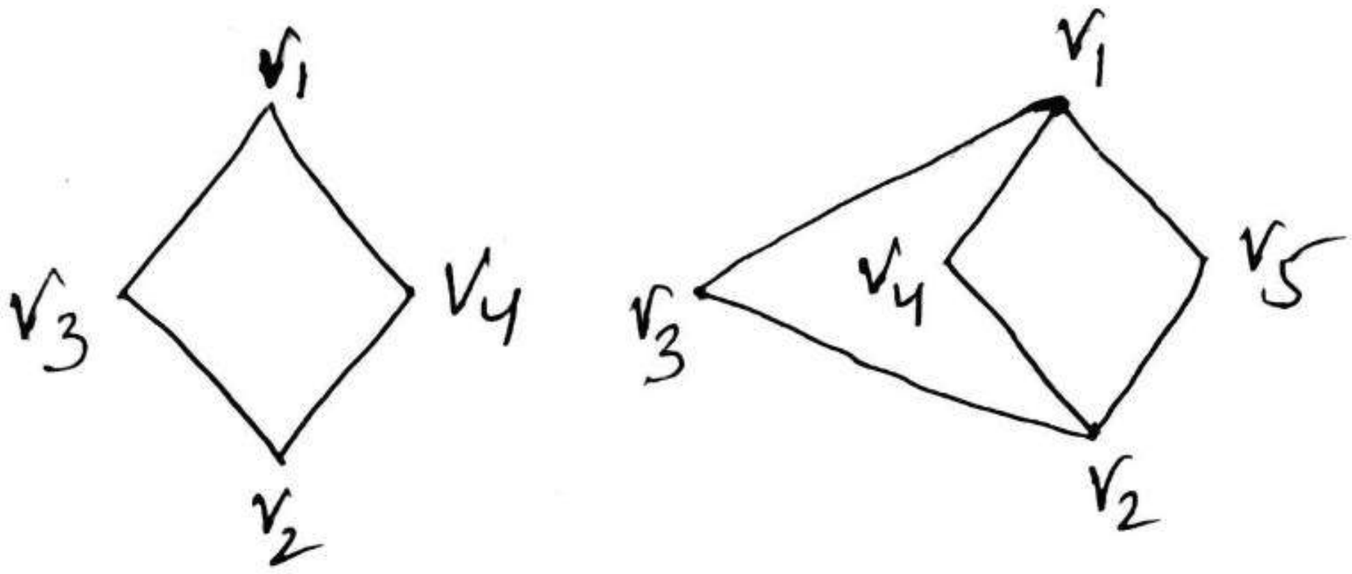
K_2 K_3 K_4

The diagrams in fig, represent the graphs

K_2, K_3, K_4. In none of these diagrams, the edges meet at points other than the vertices. Therefore K_2, K_3 and K_4 are all planar graphs.

→ S.T the bipartite graphs $K_{2,2}$ and $K_{2,3}$ are planar graphs.

Sol:



(a) $K_{2,2}$ (b) $K_{2,3}$

In $K_{2,2}$, the vertex set is made up of two bipartites V_1 and V_2, with V_1 containing two vertices say $V_1 = \{v_1, v_2\}$ and V_2 containing two vertices say $V_2 = \{v_3, v_4\}$ and there is an edge joining every vertex in V_1 with every vertex in V_2 and vice-versa. In this figure, no two edges cross each other. Therefore $K_{2,2}$ is a planar graph.

In $K_{2,3}$ the vertex set is made up of two bipartites V_1 and V_2 with V_1 containing two vertices say $V_1 = \{v_1, v_2\}$ and V_2 containing three vertices say $V_2 = \{v_3, v_4, v_5\}$ and there is an edge joining every vertex in V_1 with every vertex in V_2 and vice-versa. Fig (b) represents this graph. In this figure, no two edges cross each other. Therefore, $K_{2,3}$ is a planar graph.

→ S.T the complete graph K_5 (namely kuratowski's first graph) is non-planar.

Sol: In the complete graph K_5 there are 5 vertices and there is an edge b/w every pair of vertices, totaling to 10 edges see fig. This fig is repeated below with the vertices named as v_1, v_2, v_3, v_4, v_5 and the edges named as $e_1, e_2, e_3, \ldots e_{10}$.

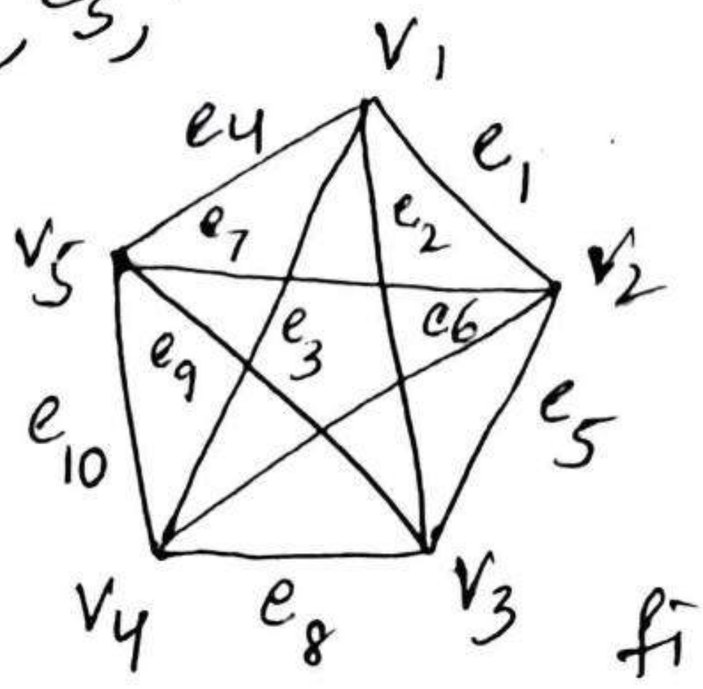


fig (a)

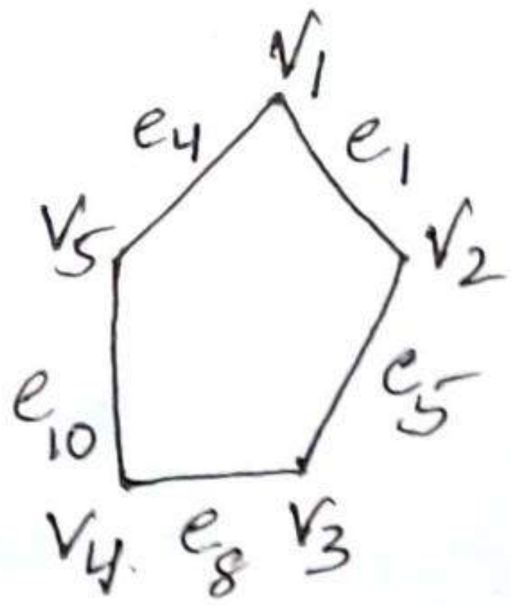
In the above drawing of K_5, the five edges $e_1, e_5, e_8, e_{10}, e_4$ form a pentagonal cycle and the remaining five edges e_2, e_3, e_6, e_7, e_9 are all inside this cycle and intersect at points other than the vertices.

Let us try to draw a diagram of K_5 in which no two of its edges intersect. The pentagonal cycle present in fig (a) does not contain any mutually intersecting edges.

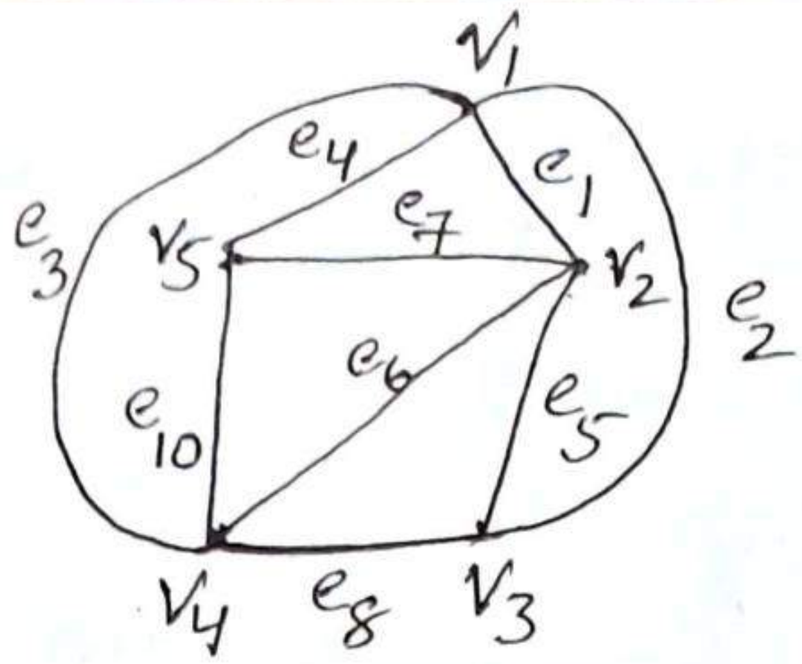
Let us start our (new) drawing of K_5 with this cycle, the cycle is shown in fig (b)

Consider the edge $e_7 = \{v_2, v_5\}$. This edge can be drawn either inside or outside the pentagonal cycle. suppose we draw it inside as shown in fig (c), the other case is similar.

Now consider the edges $e_2 = \{v_1, v_3\}$ and $e_3 = \{v_1, v_4\}$. If we draw these edges also inside the pentagon, they will intersect e_7, Therefore, let us draw both of them outside, see figure (c).



(b)



(c)

Next, consider the edge $e_6 = \{v_2, v_4\}$. If we draw this edge outside the pentagon, it intersects the edge e_2, see fig (c). Therefore, let us draw e_6 inside the pentagon.

Lastly, consider the edge $e_9 = \{v_3, v_5\}$. If we draw this edge outside the pentagon, it intersects the edge e_3, and if we draw it inside, it cross the edge e_6.

This demonstrates that in every possible plane drawing of K_5 at least two edges of K_5 intersect at a point which is not a vertex of K_5. This proves that K_5 is a <u>non-planar graph</u>.

→ <u>Thm</u> A necessary and sufficient condition for a graph G to be planar is that G does not contain K_5 or $K_{3,3}$ as a subgraph.

→ P.T the petersen graph is non-planar.

Sol: Refer Text book.

Scanned by TapScanner

<u>Def^n</u>: A graph G is said to be planar if it can be drawn on a plane without cross overs.

A graph which cannot be drawn on a plane without crossovers between its edges is called a non-planar graph.

→ A graph

Definition: <u>Region</u>: If G is a planar graph, then G can be represented by a diagram in a plane in which the edges meet only at the vertices. such a diagram divides the plane into a number of parts, called regions (or faces), of which exactly one part is unbounded. The number of edges that form the boundary of a region is called the degree of that region.

Ex: In the diagram of a planar graph shown in fig, the diagram divides the plane into 6 regions, $R_1, R_2, R_3, R_4, R_5, R_6$. we observe that each of the regions R_1 to R_5 is bounded and the

region R_6 is unbounded. i.e, R_1 to R_5 are in the interior of the graph while R_6 is in the exterior.

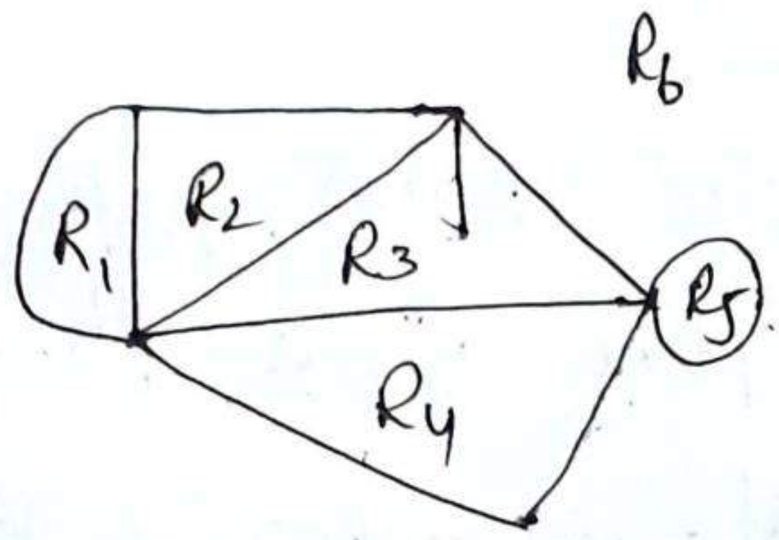


fig.

$2+3+4+3+1+6$
$19.$

We further observe that, the boundary of the region R_1 is made up of two edges.

$\therefore$ The degree of R_1 is 2

$\therefore d(R_1)=2$

The boundary of each of the regions R_2 and R_4 is made up of 3 edges.

$\therefore d(R_2)=d(R_4)=3.$

The boundary of the region R_3 consists of 4 edges of which one is a pendant edge.

$\therefore d(R_3)=$, $d(R_3)=5$

The region R_5 is bdd by a single edge (loop)

$\therefore d(R_5)=1.$

The boundary of the exterior region R_6 consists of six edges. $\therefore d(R_6)=6.$

$d(R_1)+d(R_2)+\cdots+d(R_6)=20$

which is twice the number of edges in the graph. This property is analogous to the handshaking property and is true for all planar graphs.

<u>Note</u>: If G is a connected planar graph, then any drawing of G in the plane as a plane graph will always form $|R| = |E| - |V| + 2$ regions, including the exterior region, where $|R|$, $|E|$ and $|V|$ denote the no. of regions, the no. of edges and the no. of vertices of G.

(Before Euler's this tree def^n).

Euler's Formula

<u>Thm</u>: If G is a connected plane graph, then

$$|V| - |E| + |R| = 2$$

where $|V|$ denote the no. of vertices in G, $|E|$ denote the no. of edges and $|R|$ denotes the no. of regions in G.

<u>Pf</u>. Let G is a connected plane graph.

we have to prove $|V| - |E| + |R| = 2$.

~~we prove the thm by mathematical induction on no of regions~~

Case 1: we shall prove this thm first by assuming the result for a tree.

By convention, a tree determines only one region.

We know already that the no. of edges of a

tree is one less than the number of vertices.

($\because$ n vertices $\to (n-1)$ edges)

we prove the thm by mathematical induction on no. of regions determined by G.

i.e for $n=1$. A tree is determined by only one region.

$\therefore$ we have $|E| = |V|-1$, $|R|=1$ for a tree

$\therefore |V|-|E|+|R| = |V| - \{|V|-1\} + 1 = 2$.

$\therefore$ The result holds for $n=1$. region.

we shall prove the general result by induction on the number 'n' of regions determined by G.

<u>Case II.</u> By case I, the result is true for $n=1$.

Assume that the result is true for $n \geq 1$ regions

Let G be a connected plane graph determining $(n+1)$ regions.

Delete an edge common to the boundary of two separate regions.

The resulting graph G' has the same number of vertices, one fewer edge, but also one fewer region since two previous regions

$n=2$
removing edge
$R=1$,

have been consolidated by the removal of the edge.

So if $|E'|, |V'|, |R'|$ are the no. of edges, vertices and regions respectively for G', then

$$|V'| - |E'| + |R'| = 2 \quad \text{—(1)}$$

(∵ since it is true for 'n' regions)

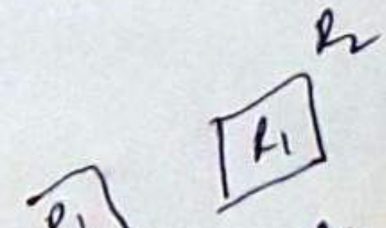
Also we have $|V'| = |V|$, $|E'| = |E| - 1$, and $|R'| = |R| - 1$

$$\therefore |V| - |E| + |R| = |V'| - [|E'| + 1] + [|R'| + 1]$$

$$= |V'| - |E'| - 1 + |R'| + 1$$

$$= |V'| - |E'| + |R'|$$

$$= 2 \quad \text{(from (1))}.$$



By mathematical induction the result holds for all connected graphs.

1) In a simple connected plane graph G, with $|E| > 1$ then

(i) $|E| \leq 3|V| - 6$

(ii) There is a vertex 'v' of G such that degree(v) ≤ 5

Pf: Let

$m \leq 4$
$|E| \leq 2|V| - 4$

DY. DSC

Note: If G is a connected simple planar graph with $(n \geq 3)$ vertices, $m(>2)$ edges and 'r' regions, then

$|E| \geq \frac{3}{2}|R|$ $\qquad m \geq \frac{3}{2}r$ $\qquad |E| \geq \frac{3}{2}|R|$

(i) $m \geq \frac{3}{2}r$ $\qquad$ (ii) $m \leq 3n-6$

Further if G is triangle-free, then

(iii) $m \leq 2n-4$.

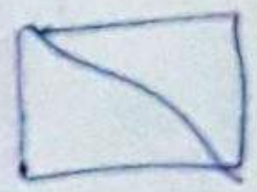
$|E| \leq 2|V|-4$

1) ~~If~~ G is a simple connected planar graph, with any $|E|>1$, then (i) $|E| \leq 3|V|-6$ ✓

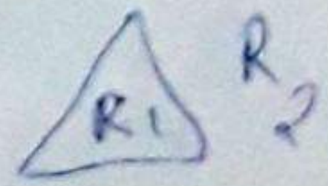
(ii) There exists a vertex v of G, such that $\deg(v) \leq 5$.

Sol: G is a simple connected graph.

$\therefore$ each region of G is bdd by atleast three edges and each edge belongs exactly two regions.



$\therefore \quad 2|E| \geq 3|R|$



$\Rightarrow \quad |R| \leq \frac{2}{3}|E|$.

$\Rightarrow \quad |V|+|R| \leq |V|+\frac{2}{3}|E|$ —— (1) (adding $|V|$ on b.s).

by Euler's formula, we have

$|V|-|E|+|R|=2$

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

substituting in the L.H.S of ①, we get

$$|E|+2 \le |V| + \frac{2}{3}|E|$$

$$\Rightarrow 3|E|+6 \le 3|V| + 2|E|$$

$$\Rightarrow 3|E| - 2|E| \le 3|V| - 6$$

$$\Rightarrow |E| \le 3|V| - 6.$$

(ii) Let each vertex of G have degree(v) ≥ 6

$$\sum_{v \in V} \deg v = 2|E| \quad \text{(by handshaking Pro.)}$$

(vertices $v = 6$.
$E = v-1 = 6-1 = 5$.
$6 \le 2|E| \le 2\times 5$

$$\Rightarrow 6|V| \le 2|E|$$

$$\Rightarrow |V| \le \frac{2}{6}|E|$$

$$\text{or } |V| \le \frac{1}{3}|E|$$

but $|R| \le \frac{2}{3}|E|$.

$$\therefore |V| + |R| \le |V| + \frac{2}{3}|E| \quad \text{(adding } |V| \text{ on b.sides)}$$

$$\Rightarrow |V| + |R| \le \frac{1}{3}|E| + \frac{2}{3}|E| \quad (\because |V| \le \frac{1}{3}|E|)$$

$$\Rightarrow |V| + |R| \le |E| \quad \text{— ①}$$

by Euler's formula,

$$|V| - |E| + |R| = 2$$

$$\Rightarrow |V| + |R| = |E| + 2 \quad \text{— ②}$$

from ① and ②, we have

$|E| + 2 \leq |E|$

$\Rightarrow 2 \leq 0$.

which is a contradiction

$\therefore$ The graph G cannot have a vertex of degree ≥ 6

Hence, each vertex of G is degree ≤ 5.

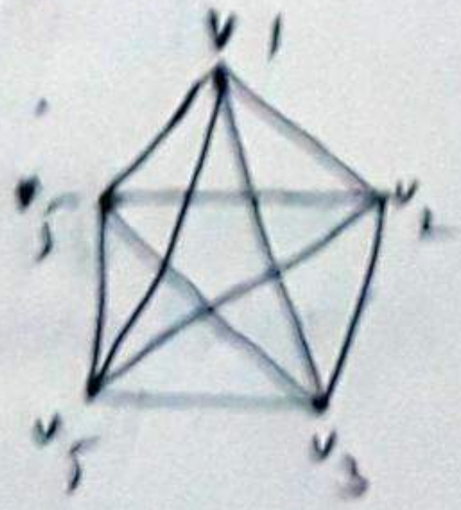
2) P.T K_5 is non-planar.

Sol: The graph K_5 is simple, connected and has no. of vertices $|V| = 5$, no. of edges in $K_5 = |E| = 10$.

If G is planar then $|E| \leq 3|V| - 6$

$10 \leq 3(5) - 6$

$10 \leq 9$



which is a contradiction

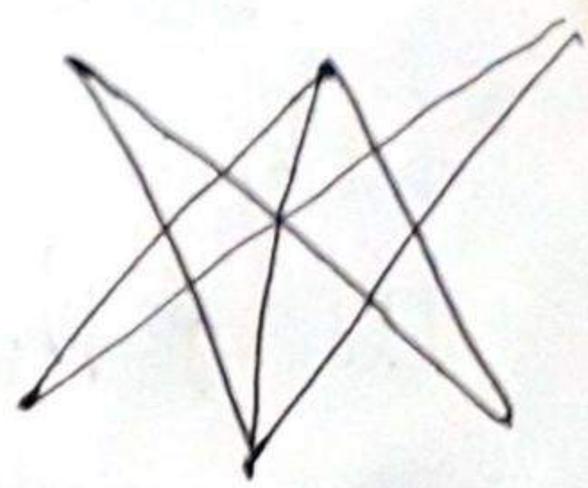
$\therefore K_5$ is non-planar.

3) Prove $K_{3,3}$ is non-planar. $|E| \leq 2|V| - 4$

Pf: we first note that Kuratowski's second graph $K_{3,3}$ is simple, connected and has $|V| = 6$ and $|E| = 9$ edges

Suppose $K_{3,3}$ is planar,

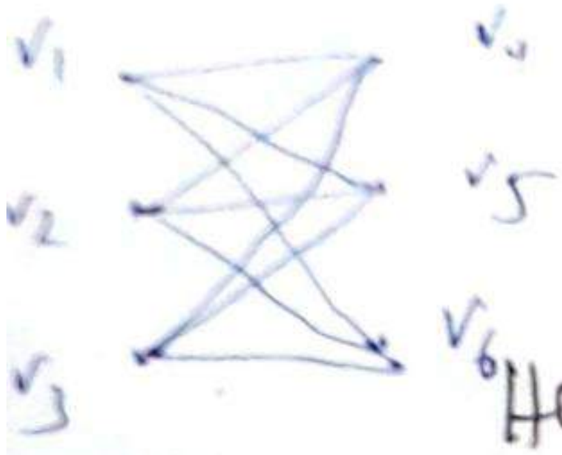
we note that $K_{3,3}$ has no triangle -



$\therefore$ we have $m \le 2n-4$

i.e $|E| \le 2|V|-4$

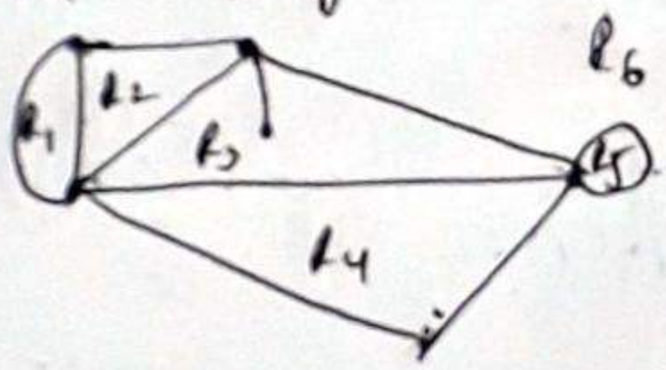
$9 \le 2(6)-4$



$9 \le 8$ which is not true.

Hence $K_{3,3}$ is non planar.

4) verify Euler's formula for the planar graph shown in fig.



Sol: The given graph has $|V| = 6$ vertices, $|E| = 10$

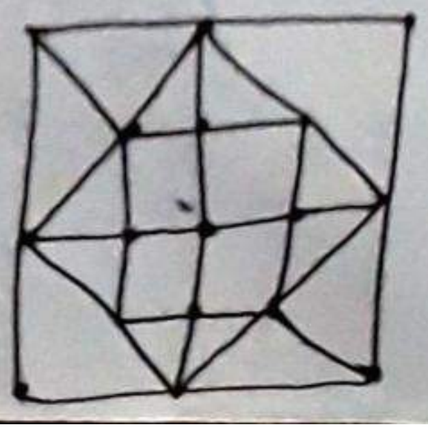
$|R| = 6$ regions

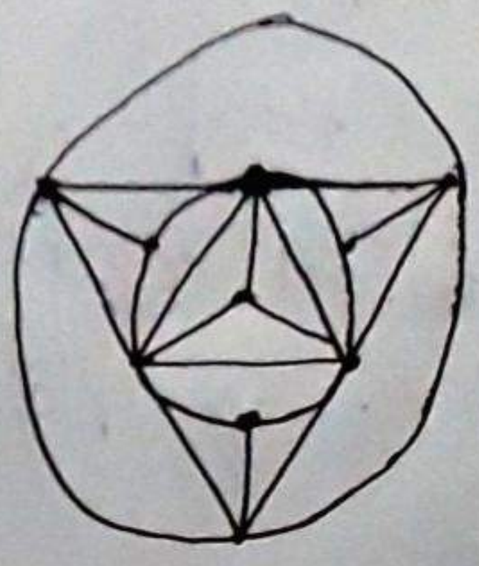
By Euler's formula,

$|V| - |E| + |R| = 6 - 10 + 6 = 2$

The Euler's formula is verified for the given graph.

5) verify Euler's formula for the planar graphs shown below.





we observe that the first of the given graphs has $|V| = 17$ vertices, $|E| = 34$ edges and $|R| = 19$

Thus, By Euler's formula

$$|V| - |E| + |R| = 17 - 34 + 19 = 2.$$

(ii) In the second of the given graphs, there are $|V| = 10$, $|E| = 24$, and $|R| = 16$

$$\therefore |V| - |E| + |R| = 10 - 24 + 16 = 2$$

$\therefore$ Euler's formula is verified.

Dual of a planar graph:

Consider a connected planar graph G. Suppose R_1, R_2, R_3, etc, be the regions (including the exterior region) in the drawing. Let us now construct a graph G^* following the procedure given below:

1) Choose one point inside each of the regions $R_1, R_2, R_3 \ldots$ Denote these points by $v_1^*, v_2^*, v_3^*, \ldots$ These points will be the vertices of G^*.

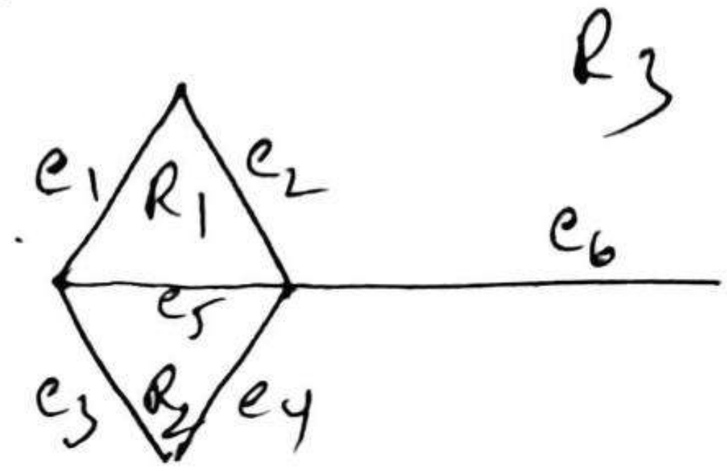
2) If two regions R_i and R_j are adjacent (i.e have a common edge say e_k) draw a line e_k^* joining the points v_i^* and v_j^* that intersects the common edge e_k exactly once.

3) If there is more than one edge common to R_i and R_j, draw one line e_p^* b/w the points v_i^* and v_j^* for each common edge e_p, intersecting e_p exactly once.

4) For an edge e_i lying entirely in one region say R_i, draw a loop e_i^* at the point v_i^* intersecting e_i exactly once.

The graph G^* so constructed is called a geometric dual or just a dual of G.

Ex: consider the graph G shown in fig



we observe that the figure divides the plane into three regions R_1, R_2 & R_3 of which R_3 is unbdd.

we construct the dual G^* of G by using the above stated procedure.

i) we choose three points v_1^*, v_2^*, v_3^* inside the regions R_1, R_2, R_3 respectively

(ii) The regions R_1 and R_2 have a common edge e_5. we draw a line e_5^* joining v_1^* and v_2^* that crosses e_5 exactly once.

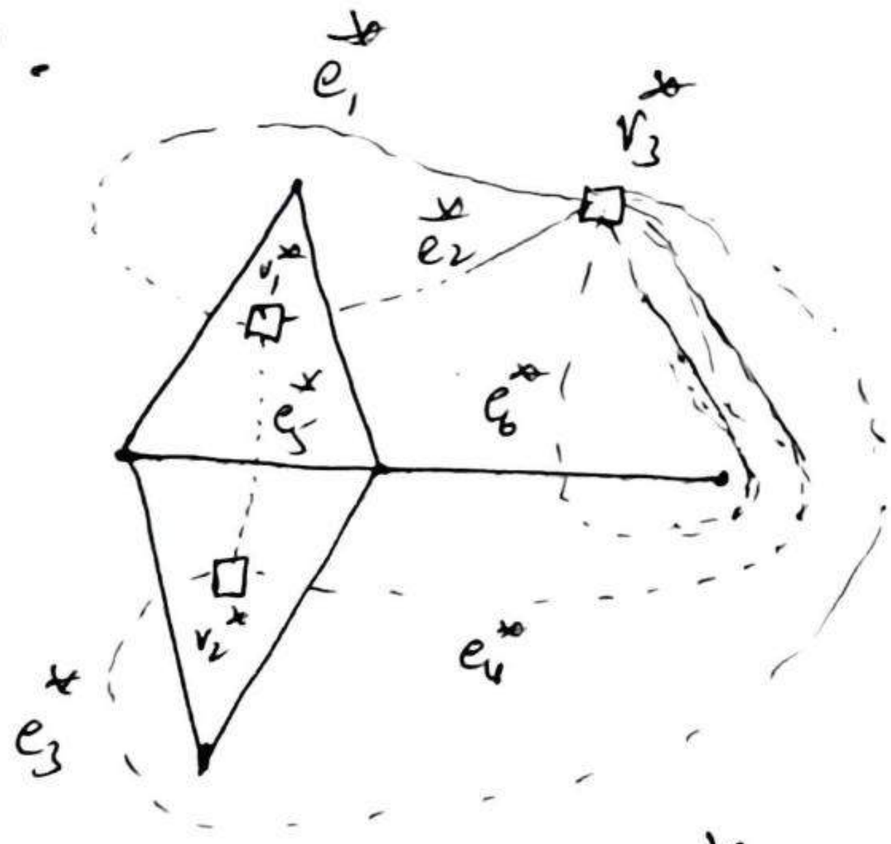
(iii) The regions R_1 and R_3 have two common edges e_1, e_2. we draw two lines e_1^*, e_2^* b/w v_1^* and v_3^*, with e_1^* crossing only e_1 and e_2^* crossing only e_2.

(iv) The regions R_2 and R_3 have two common edges e_3, e_4. we draw two lines e_3^*, e_4^* b/w v_2^* and v_3^* with e_3^* crossing only e_3 and e_4^* crossing only e_4.

Scanned by TapScanner

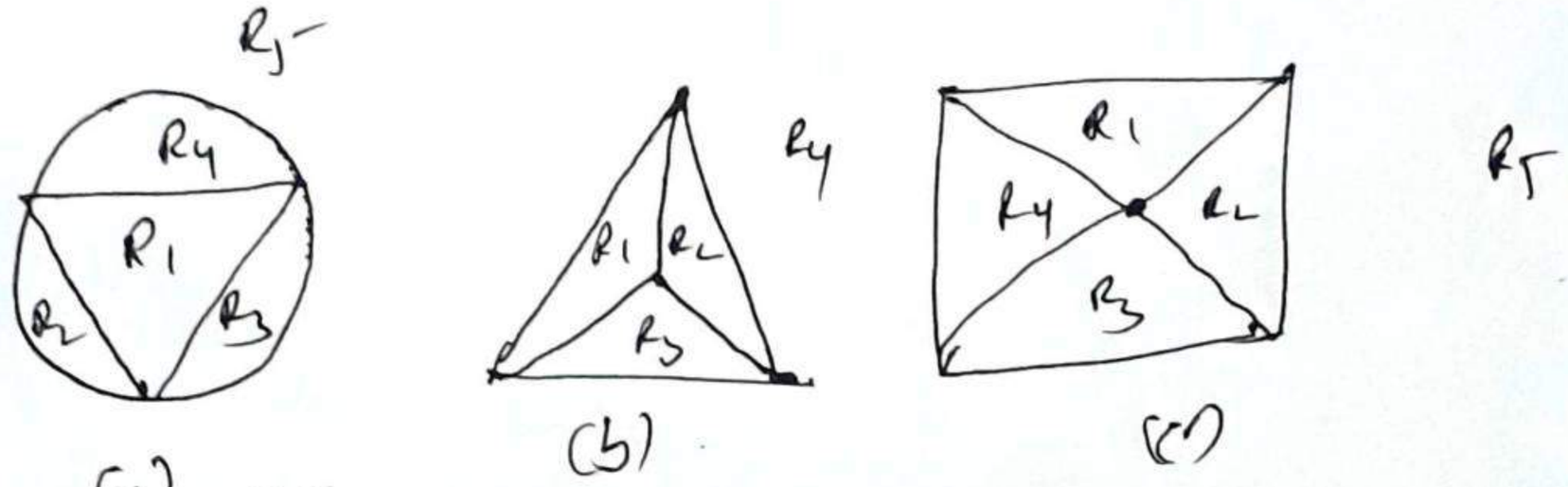
(v) The edge e_6 is completely contained in R_3. we draw a loop e_6^* at v_3^* intersecting e_6 exactly once.

The construction of G^* is now complete. For G^*, the pts v_1^*, v_2^*, v_3^* are the vertices the lines $e_1^*, e_2^*, e_3^*, e_4^*, e_5^*$, and e_6^* are the edges.

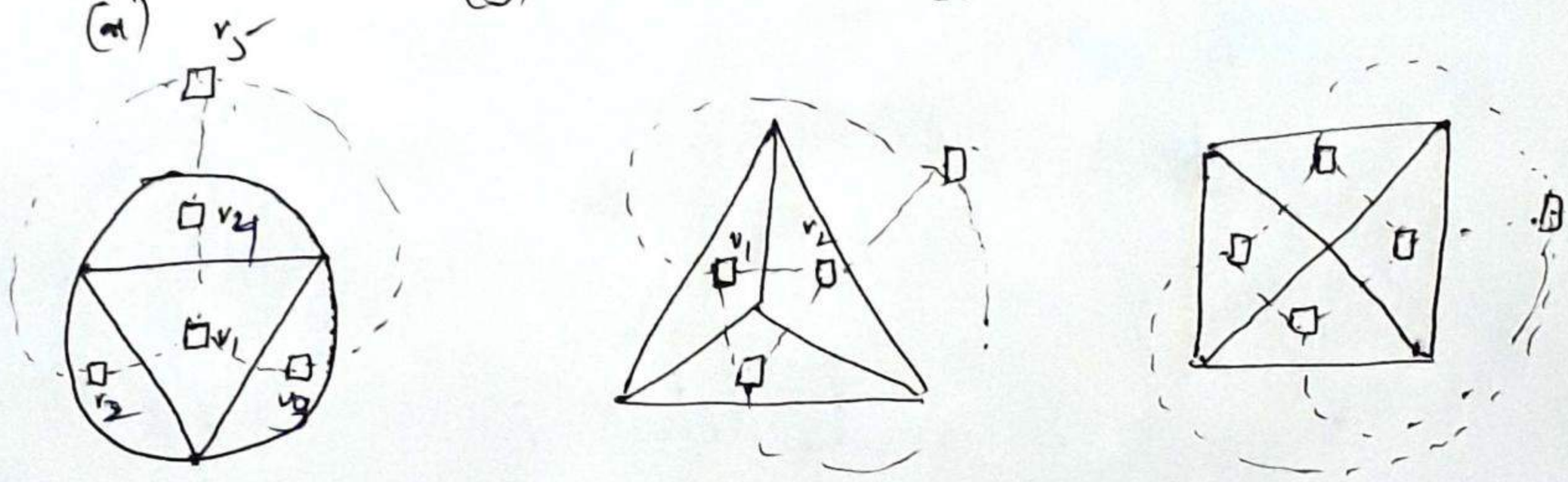


Both of G & G^* are shown in fig. The vertices of G^* are indicated by small squares and its edges by dotted lines.

1) construct the duals of the foll. planar graphs.



Sol:



Graph coloring

Some Results

The result listed below follow from the procedure used in the construction of G^*:

1) There is a one-to-one correspondence b/w the regions of G and the vertices of G^*.

2) There is a one-to-one correspondence b/w the edges of G and the edges of G^*. (If e^* is the edge of G^* that corresponds to the edge e of G, then e^* is called the dual of e).

3) A pendant edge in G yields a loop in G^*.

4) A loop in G yields a pendant edge in G^*.

5) Edges that are in series in G yield parallel edges in G^*.

6) Parallel edges in G yield edges in series in G^*.

7) The no. of edges which form a boundary of a region R_i in G is equal to the degree of the corresponding vertex v_i^* in G^*, and vice-versa.

8) Like G, G^* is a connected planar graph.

9) If G is a connected planar graph with 'n' vertices, m edges and 'r' regions. and its dual G^* has n^* vertices, m^* edges and r^* regions, then

$$n^* = r, \quad m^* = m, \quad r^* = n.$$

→ <u>Self Dual Graphs</u>: Let G be a planar graph and G^* denote the dual of G. If G and G^* are isomorphic to each other then G is called a self dual graph.

Scanned by TapScanner

Graph colouring:

$|E| \leq 3|V| - 6$

$|E| \leq 2|V| - 4$

Given a planar or non-planar graph G, If we assign colours to its vertices in such a way that no two adjacent vertices have (receive) the same colour, then we say that the graph G is properly coloured.

In other words, proper colouring of a graph means assigning colours to its vertices such that adjacent vertices have different colours.

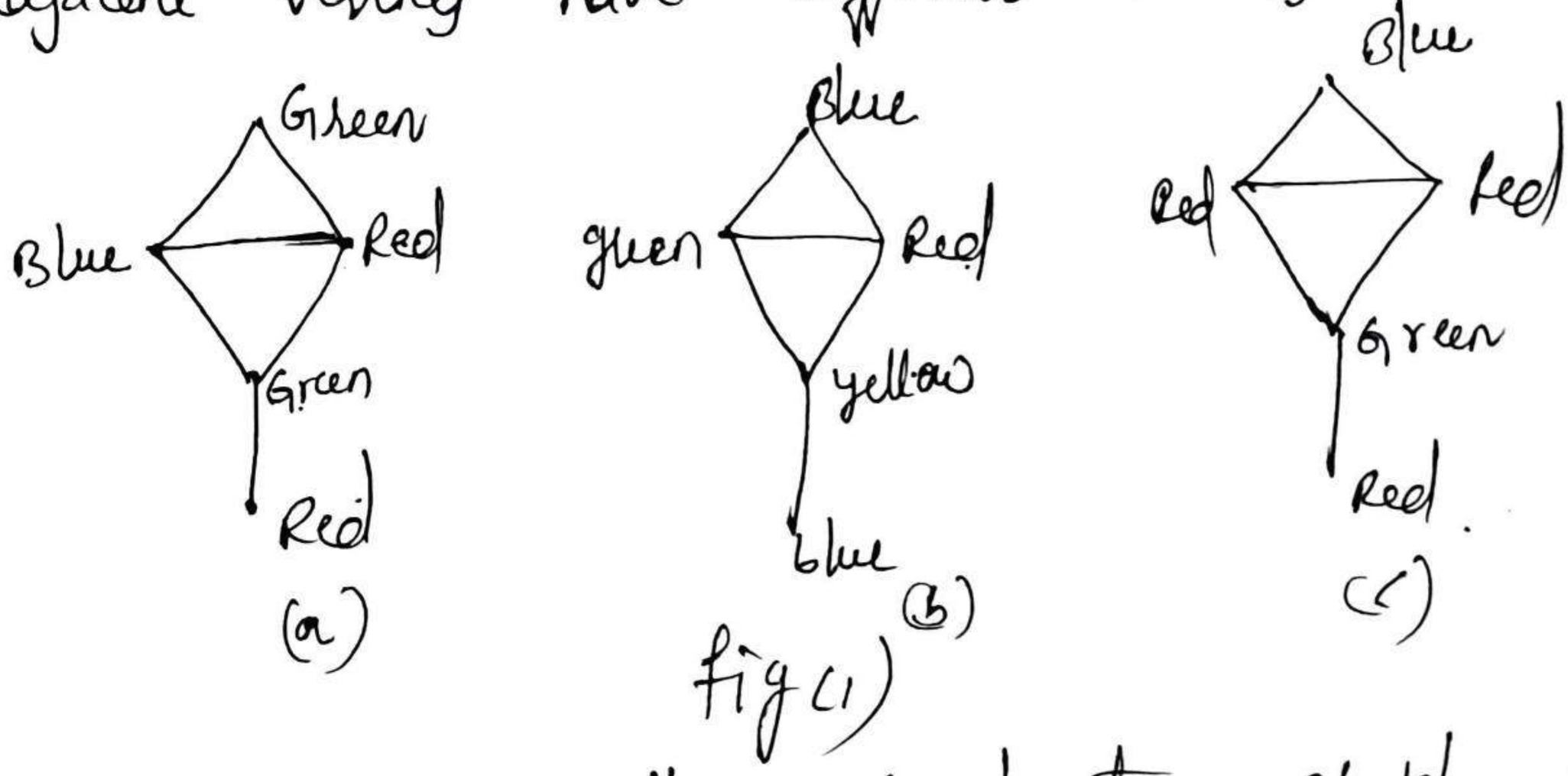


fig (1)

In figure (1), the first two graphs are properly coloured whereas the third graph is not properly coloured.

By examining the first two graphs in fig (1)

which are properly coloured, we note the following

1) A graph can have more than one proper colouring.

2) Two non-adjacent vertices in a properly coloured graph can have the same colour.

chromatic number:

→ A graph G is said to be k-colourable if we can properly colour it with k (number of) colours.

→ A graph G which is k-colourable but not (k-1) colourable is called a k-chromatic graph.

In otherwords, a k-chromatic graph is a graph that can be properly coloured with k colours but not with less than k colours.

If a graph G is k-chromatic, then k is called the chromatic number of G.

Thus, the chromatic number of a graph is the minimum number of colours with which the graph can be properly coloured.

The chromatic number of a graph G is usually denoted by $\chi(G)$.

Scanned by TapScanner

Def^n: The chromatic number of a graph G, is the minimum number of colours required to colour the vertices of G, such that no two adjacent vertices receive the same colour.

Some Results

The following results are direct consequences of the def^n of the chromatic number.

1) A graph consisting of only isolated vertices (i.e. Null graph) is 1-chromatic (Because no two vertices of such a graph are adjacent and therefore we can assign the same colour to all vertices)

2) A graph with one or more edges is at least 2-chromatic. (Because, such a graph has at least one pair of adjacent vertices which should have different colours).

3) If a graph G contains a graph G_1 as a subgraph, then $\chi(G) \geq \chi(G_1)$.

Scanned by TapScanner

4) If G is a graph of n vertices, then

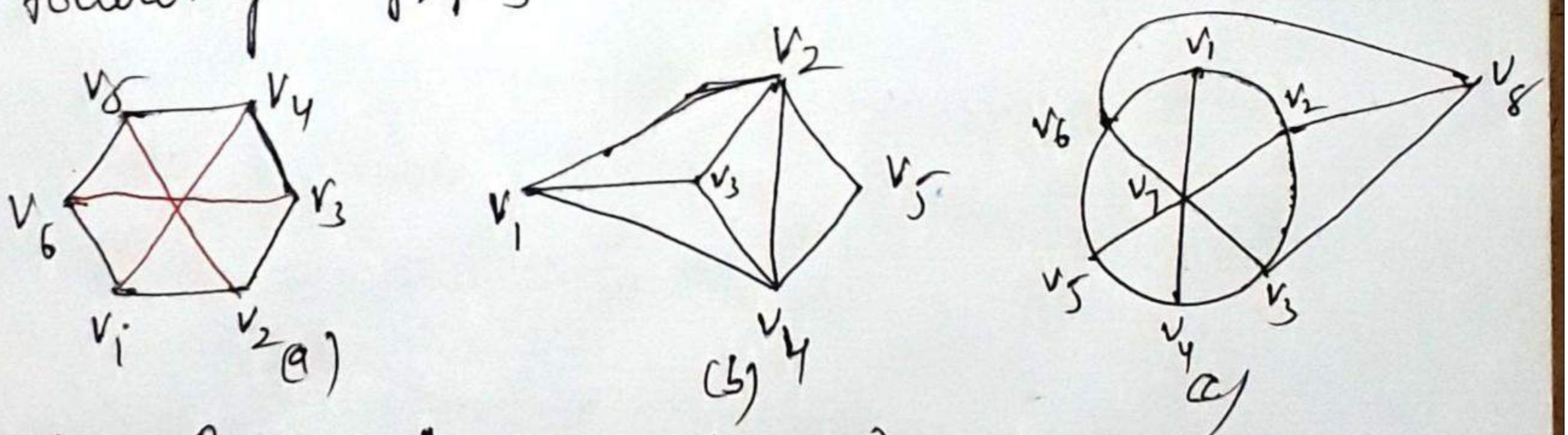
$$\chi(G) \leq n.$$

5) $\chi(K_n) = n$ for all $n \geq 1$ (Because, in K_n, every two vertices are adjacent, and as such all the 'n' vertices should have different colours).

6) If a graph G contains K_n as a subgraph, then

$$\chi(G) \geq n.$$

<u>Problems</u>: Find the chromatic number of each of the following graphs



<u>Sol</u>: (i) For the graph (a), let us assign a colour α to the vertex v_1. Then, for a proper colouring, we have to assign a different colour to its neighbours v_2, v_4, v_6. Since v_2, v_4, v_6 are mutually non-adjacent vertices, they can have the same colour, say β (which is different from α).

Since v_3, v_5 are not adjacent to v_1, these can have the same colour as v_1, namely α.

Thus, the graph can be properly coloured with at least two colours, with the vertices v_1, v_3, v_5 having one colour α, and v_2, v_4, v_6 having a different colour β. Hence the chromatic no. of the graph is 2.

(ii) For the graph (b), let us assign the colour 'α' to the vertex v_1. Then for a proper colouring, its neighbours v_2, v_3 and v_4 cannot have the colour α, but v_5 can have the colour α.

Furthermore v_2, v_3, v_4 must have different colours say β, γ, δ. Thus at least four colours are required for a proper colouring of the graph. Hence the chromatic no. of the graph is 4.

iii) For the graph (c), we can assign the same colour say α to the non-adjacent vertices v_1, v_3, v_5. Then the vertices v_2, v_4, v_6 can be assigned

the same colour other than α. Suppose we assign a colour β to v_2, v_4, v_6. Consequently v_7 and v_8 can be assigned the same colour which is different from both α and β.

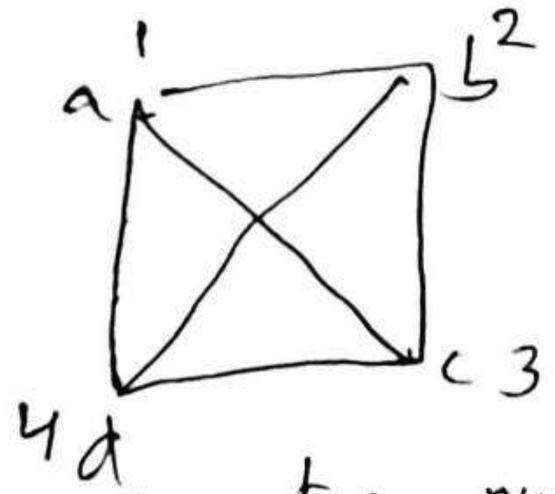
Thus, a minimum of three colours are needed for a proper colouring of the graph. Hence its chromatic number is 3.

2) what is the chromatic number of a K_n?

Sol: In K_n, every vertex is adjacent to all the remaining $(n-1)$ vertices. So every vertex we need a different colour.

$\therefore$ The chromatic no. for K_n is $\chi(K_n) = n$.

Ex:

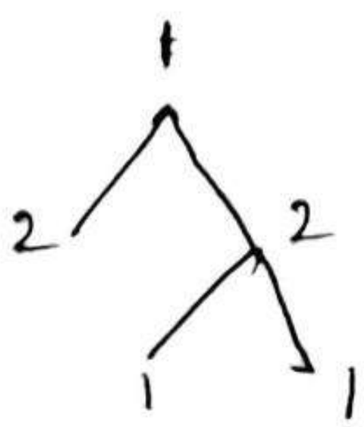


For K_4, The minimum no. of colour required for colouring K_4 is $\chi(K_4) = 4$.

→ what is the chromatic number of a Tree?

Sol: The chromatic number of a non-trivial tree is 2 and it is 1 for a trivial tree.

Trivial Tree



Non-Trivial tree

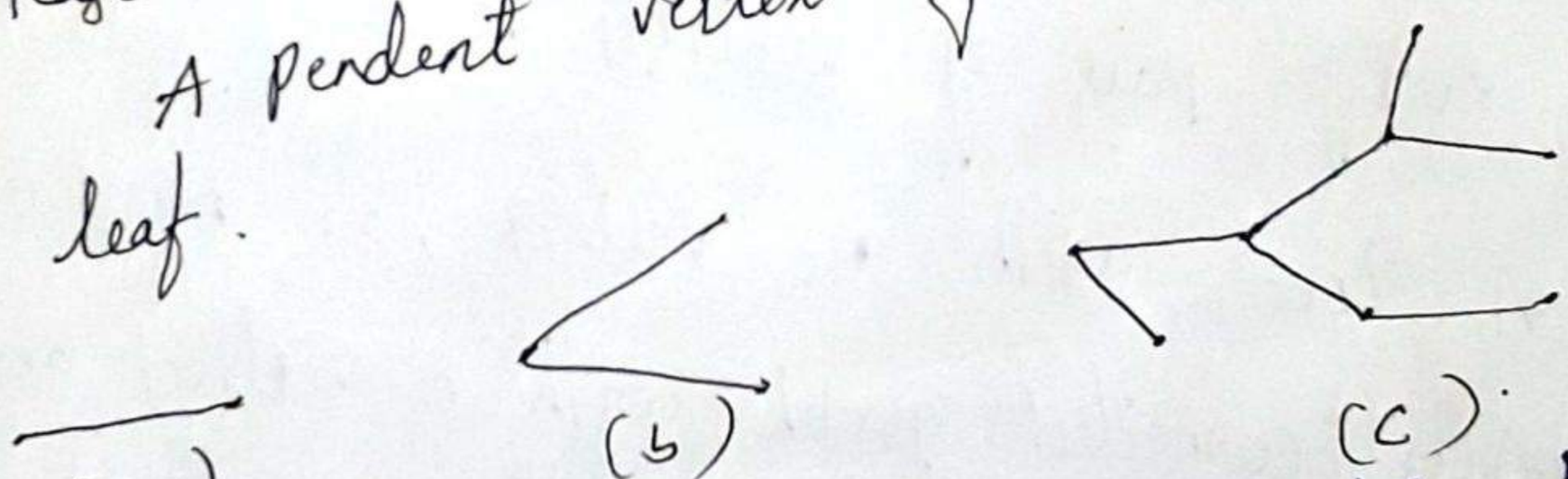
Trees and their Basic Properties

A graph G is called a tree if it is connected and has no cycles.

It immediately follows that a tree has to be a simple graph, because loops and parallel edges form cycles.

The graphs shown in fig(1) are all trees. we observe that each of these trees possesses at least two pendant vertices.

A pendent vertex of a tree is also called a leaf.

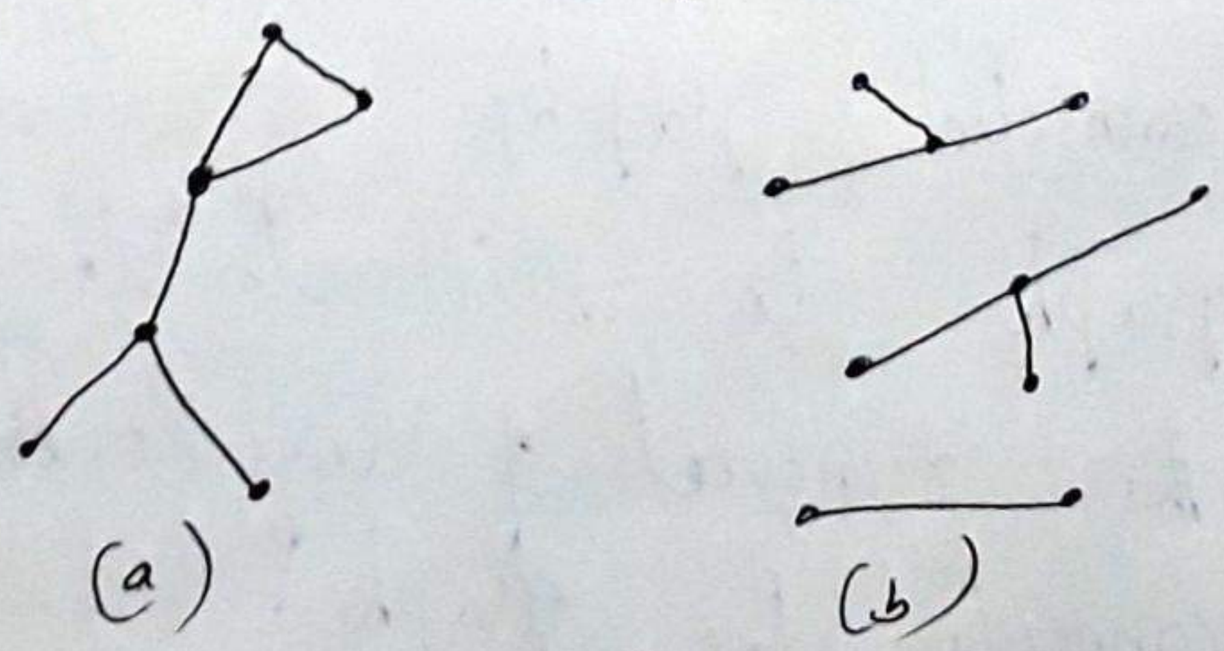


(a) (b) (c)

fig (1)

Trivial & non-trivial Trees:

A tree which consists of single vertex is called trivial tree otherwise it is called as non-trivial tree.



(a) (b)

fig (2)

The graphs shown in fig (2) are not trees.

observe that the first of these contains a cycle whereas the second is not connected. However, each component of the second (disconnected) graph is a tree. such a graph is called a forest.

→ A graph which is a tree is usually denoted by T (instead of G) to emphasize the structure.

Properties of trees.

Thm: 1) In a tree, there is one and only one path b/w every pair of vertices.

Thm2: A tree with n vertices has n-1 edges.

Thm3: Any connected graph with 'n' vertices and (n-1) edges is a tree.

Thm4: Minimally connected graphs:

A connected graph is said to be minimally connected if the removal of any one edge from it disconnects the graph.

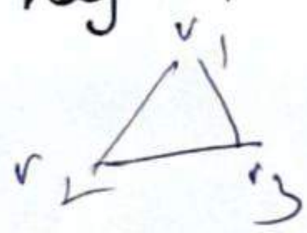
Thm A connected graph is a tree if and only if it is minimally connected.

Prob

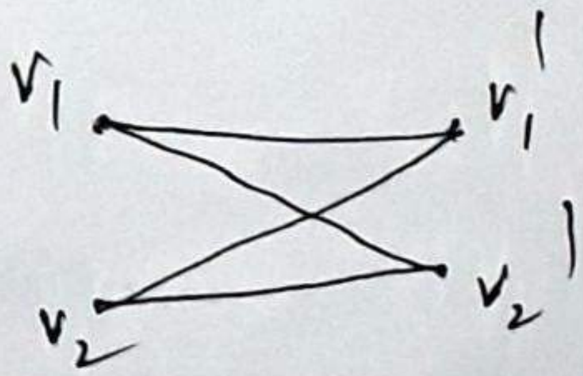
1) a) S.T the complete graph K_n is not a tree when $n > 2$.

b) S.T the complete bipartite graph $K_{r,s}$ is not a tree when $r \geq 2$.

Sol: a) If v_1, v_2, v_3 are any three vertices of K_n, $n>2$, Then the closed walk $v_1 v_2 v_3 v_1$ is a cycle in K_n. Since K_n has a cycle, it cannot be a tree.

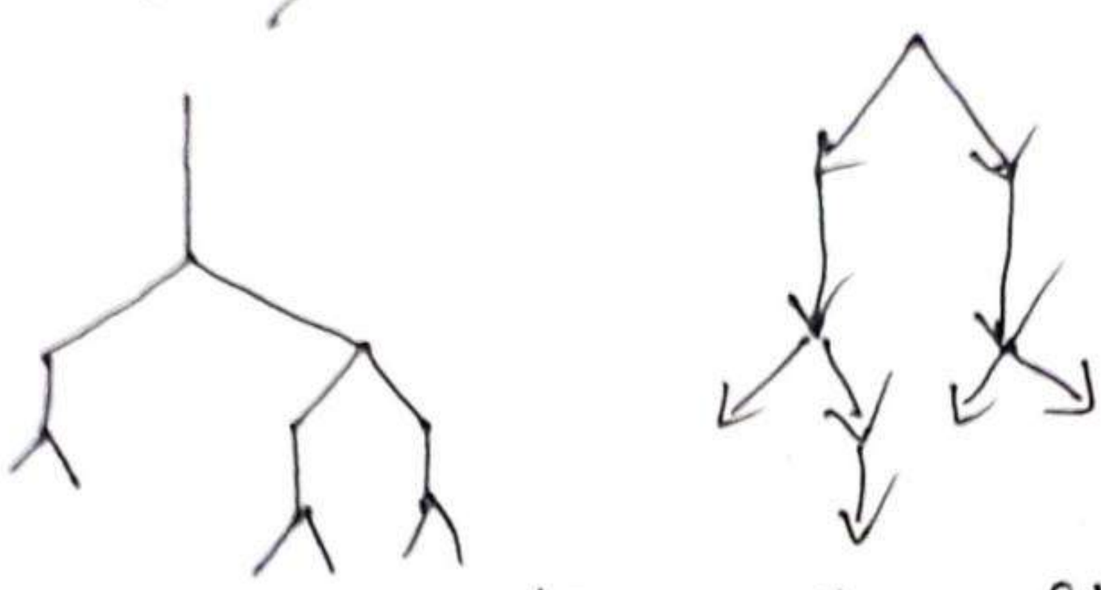


b) Let v_1 and v_2 be any two vertices in the first bipartite and v_1', v_2' be any two vertices in the other bipartite of $K_{r,s}$, $s \geq r > 1$. Then the closed walk $v_1 v_1' v_2 v_2' v_1$ is a cycle in $K_{r,s}$. Since $K_{r,s}$ has a cycle, it cannot be a tree.



Defn: <u>Directed Tree</u>: A connected, acyclic directed graph is called a directed tree.

The graph (a) is a non-directed tree and the graph (b) is directed tree.



<u>Note:</u> 1) If T is a tree, then it has a unique simple non-directed path b/w each pair of vertices.

2) A tree with only one vertex is called a Trivial tree. If T is not trivial tree then it is called a non-trivial tree.

Theorem <u>Thm</u> A graph G is a tree if and only if G has no cycles and $|E| = |V| - 1$.

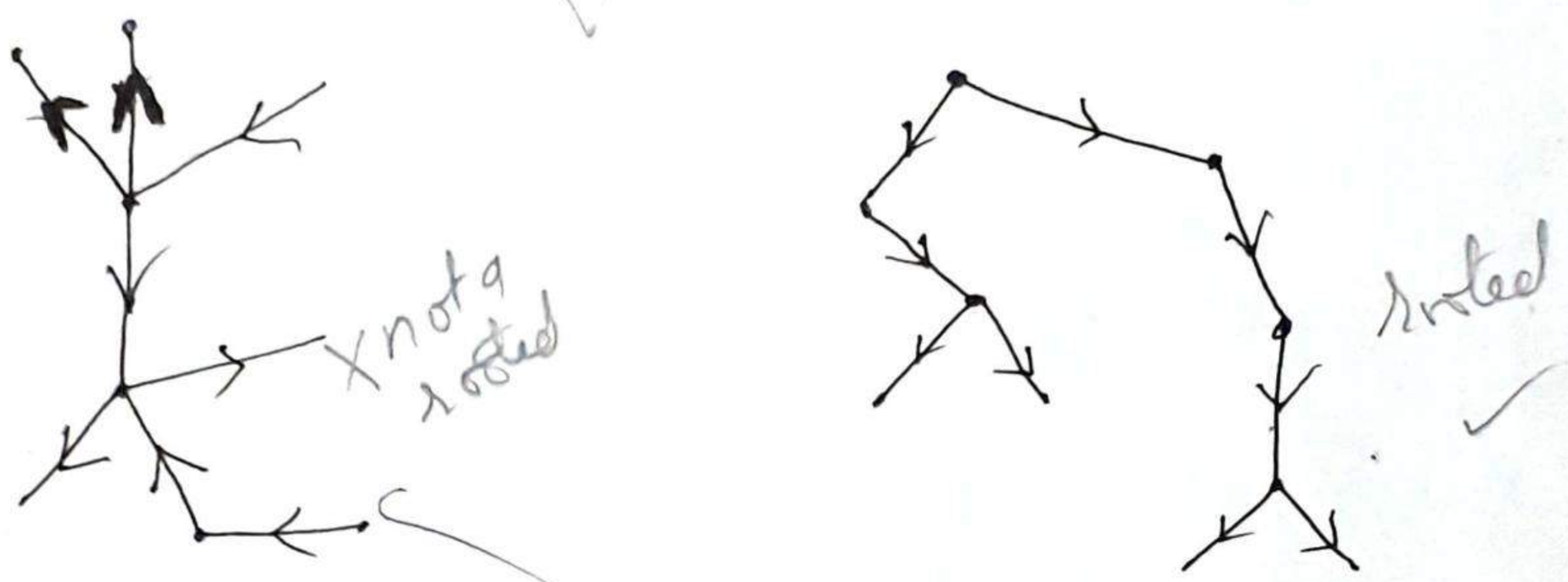
<u>Rooted Trees</u>: A directed tree T is called a rooted tree if

(i) T contains a unique vertex, called the root, whose in-degree is equal to '0' and

(ii) the <u>in-degrees</u> of all other vertices of T are <u>equal to</u> 1.

figures (a) & (b) depict two directed Trees. The first of these is not a rooted Tree whereas

the second is a rooted tree.

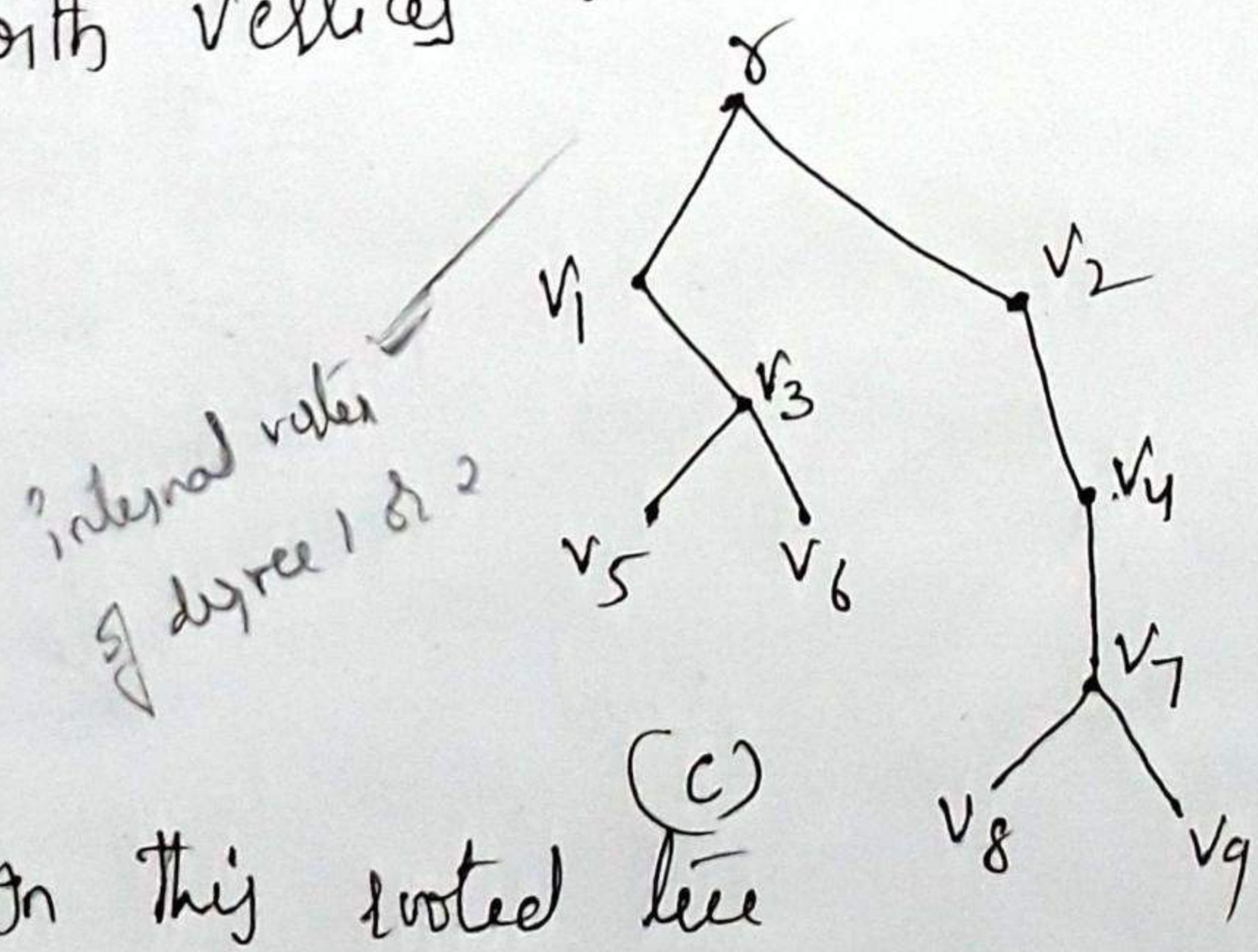


(a) fig (1) (b)

For example, suppose we redraw the directed tree of fig (b) as, without arrows and with vertices labeled.

for v_2 → of out degree = 1
v_4 → " = 1
v_7 = out deg = 2

binary tree



I (c)

(c)

In this rooted tree

1) v_1 and v_2 are at the first level, v_3, v_4 are at the second level, v_5, v_6, v_7 are at the third level and v_8 & v_9 are at the fourth level.

2) v_1 is the ancestor of v_3, v_5, v_6 (or v_3, v_5, v_6 are the descendants of v_1), and v_2 is the ancestor of v_4, v_7, v_8, v_9 (or v_4, v_7, v_8, v_9 are the descendants of v_2).

3) v_1 is the parent of v_3 (or v_3 is a child of v_1)

4) v_5 and v_6 are siblings, and v_8, v_9 are siblings.

5) v_5, v_6, v_8, v_9 are leaves, and all other vertices are internal vertices.

<u>Binary Tree</u>: A rooted tree T is called a binary rooted tree or just a binary tree if every internal vertex of T is of out-degree 1 or 2.

i.e, if every vertex has at most two children.

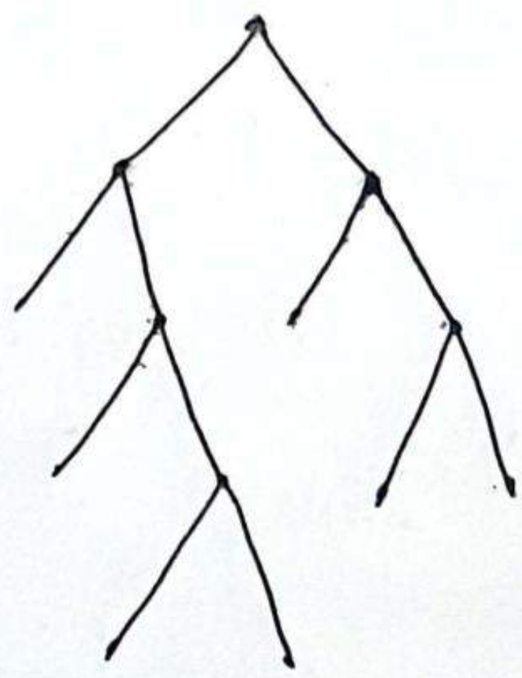
→ A rooted tree T is called a complete binary tree if every internal vertex of T is of out-degree 2,

i.e if every internal vertex has two children.

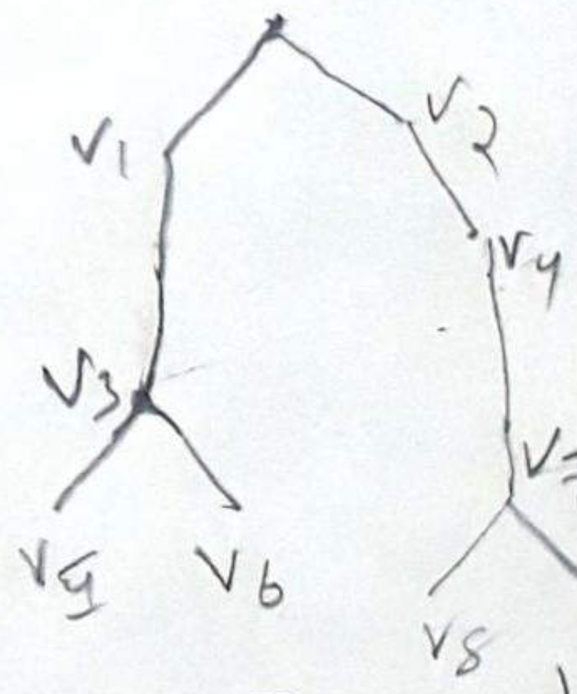
→

The rooted tree shown in figure (c) is a binary tree, it is not a complete binary Tree.

The rooted tree shown in fig(2) below is a complete binary tree.



fig(2)



v_1, v_2 - I

v_3, v_4 - II

v_5, v_6, v_7 - III

v_8, v_9 - IV

Complete Binary Tree

Binary tree

Balanced Tree: If T is a rooted tree and h is the largest level number achieved by a leaf of T, then T is said to have height h.

→ A rooted tree of height h is said to be balanced if the level number of every leaf is h or h-1

The tree shown in fig 1(c) is of height 4 and is balanced too. The tree shown in fig (2) is of height 4 but is not balanced

Full binary Tree :

Let T be a complete binary tree of height h. Then T is called a full binary tree if all the leaves in T are at level h.

The complete binary tree shown in fig (2) is not a full binary tree. The tree shown in fig (3) is a full binary tree

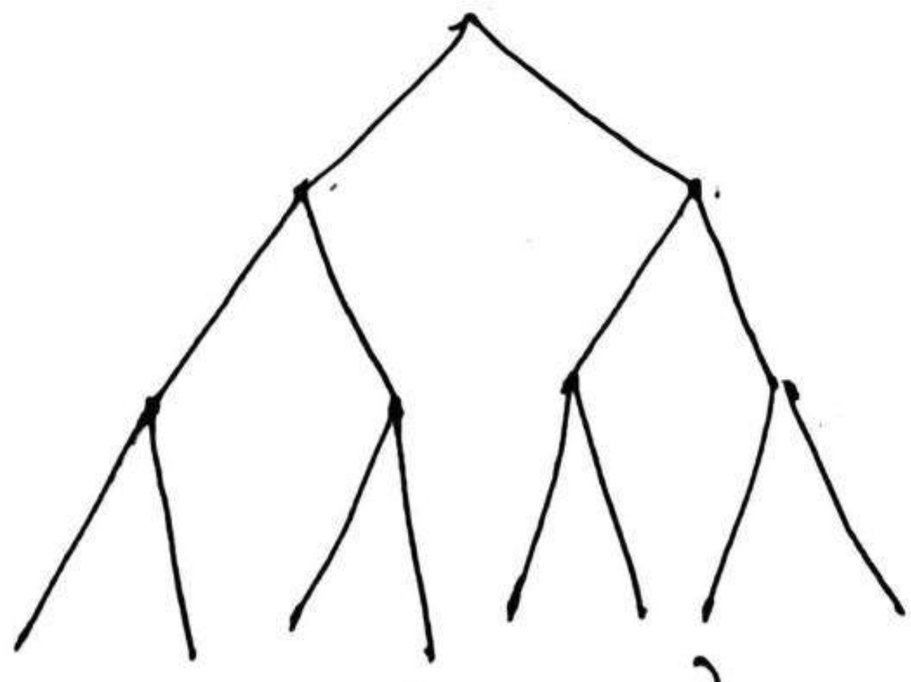


fig (3)

Scanned by TapScanner

Spanning Trees :

Let G be a connected graph. A subgraph T of G is called a spanning tree of G if

(i) T is a tree and

(ii) T contains all vertices of G.

i.e A spanning Tree of a connected graph is a spanning subgraph that is also a tree.

→ The names skeleton and scaffolding are also used for a spanning tree.

→ The edges of a spanning tree are called its branches.

Chord: If T is a spanning tree of a graph G, then the edges of G that are not in T are called the chords of G w.r.t to T.

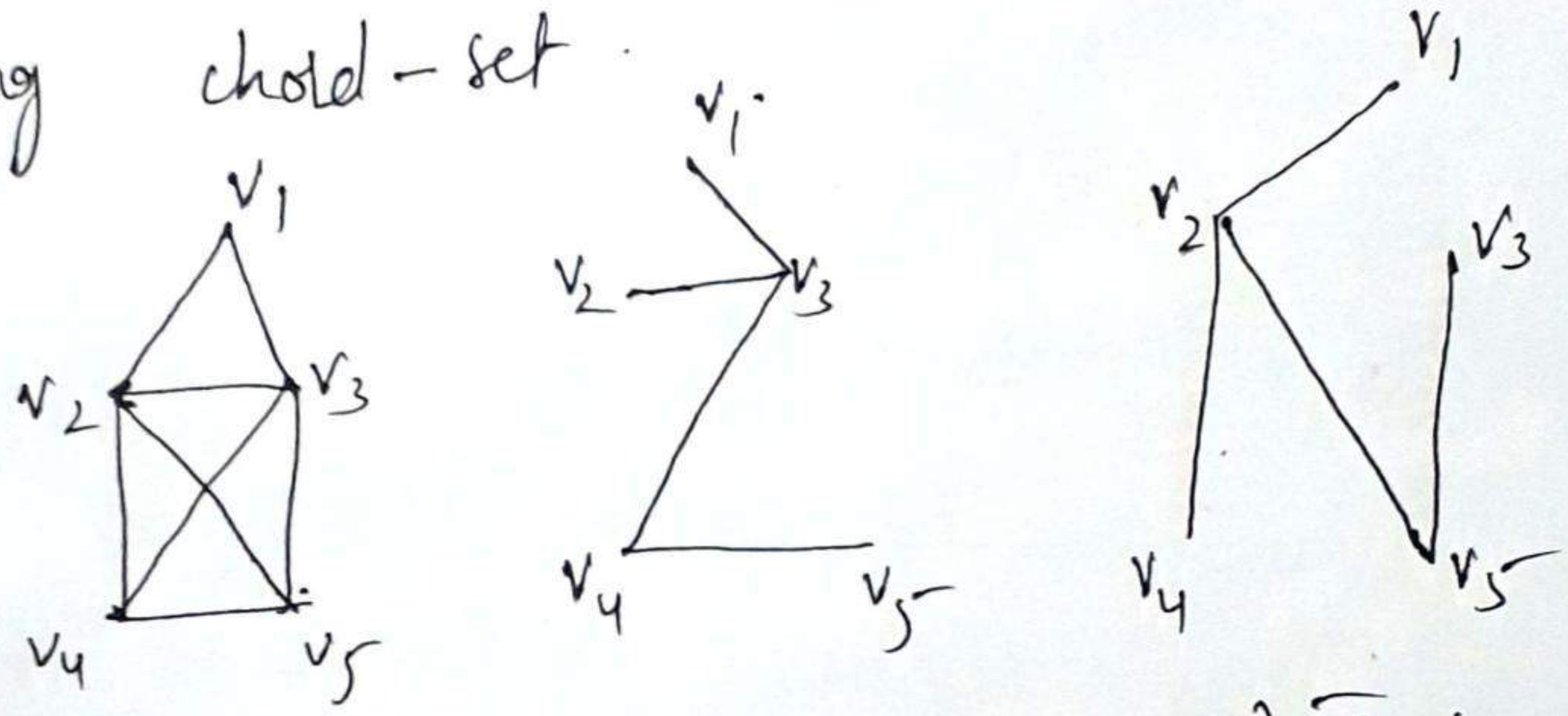
The set of all chords of G is the complement of T in G. This set is called the chord-set

Scanned by TapScanner

or cotree of T in G and is denoted by $\bar{T}$.
by $\bar{T}$

Evidently $G = T \cup \bar{T}$.

For the graph in fig (a), (b) is a spanning tree and the graph in fig (c) is the corresponding chord-set.



(a) G b): T c) $\bar{T}$

Depth-First search and Breadth-First search

we define two algorithms for finding a spanning tree of a connected graph.

The first of these is known as the depth-first-search (DFS) and the second is known as the breadth-first search (BFS)

DFS Algorithm :

Let $G = (V, E)$ be a connected graph of order n, with vertices labeled $v_1, v_2, \ldots v_n$ in some specified order.

The DFS algorithm specifies the following steps for determining a spanning tree T of G wherein the variable v stands for the vertex being considered.

Step 1: Assign the first vertex v_1 to the variable v and initialize T as the tree consisting of just this vertex. (The vertex v_1 will turn out to be the root of T that develops).

Step 2: select the smallest subscript k for $2 \le k \le n$ such that $\{v, v_k\} \in E$ and v_k has not already been included in T.

If no such subscript is found, then go to step 3. otherwise, perform the following:

(i) Attach the edge $\{v, v_k\}$ to T

(ii) Assign v_k to v

(iii) Return to step 2.

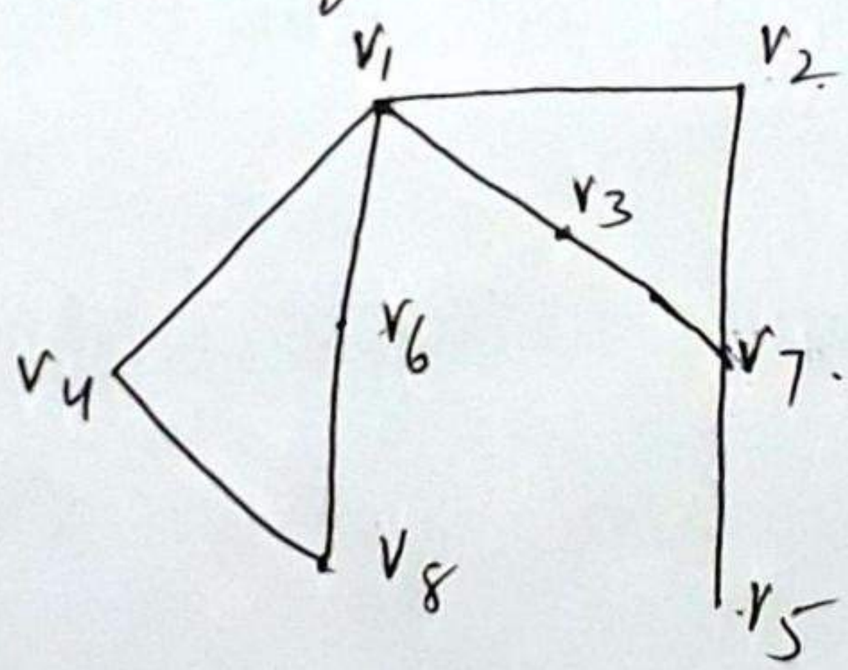
Scanned by TapScanner

Step 3: If $v = v_1$, the tree T is the spanning tree for the order specified.

Step 4: For $v \neq v_1$, back track from v. If u is the parent of the vertex assigned to v in T, then assign u to v and return to step 2.

The spanning tree of G determined by using the above steps is called the depth-first search spanning tree for G for the given order of vertices of G.

Ex: Determining a spanning tree by using DFS Algorithm



Sol: First we assign the first vertex v_1 to the variable v and initialise T as just the vertex v_1 (the root).

Going to step 2, we find that the vertex v_2 is the first vertex w such that $\{v_1, w\} \in E$ and w has not been visited earlier. So we

- attach the edge $\{v_1, v_2\}$ to T, assign v_2 to v and return the step 2.

At $v = v_2$, we find that the first adjacent vertex (not visited earlier) that provides an edge for T is v_7. Consequently the edge $\{v_2, v_7\}$ is attached to T,

v_7 is assigned to v, and we again return to step 2.

At $v = v_7$, we find that the first adjacent vertex (not visited earlier) that provides an edge for T is v_3. Consequently, the edge $\{v_7, v_3\}$ is attached to T, v_3 is assigned to v and we return to step 2.

Now we note that there is no new vertex which we can obtain from v_3, because the adjacent vertex v_1 has already been visited.

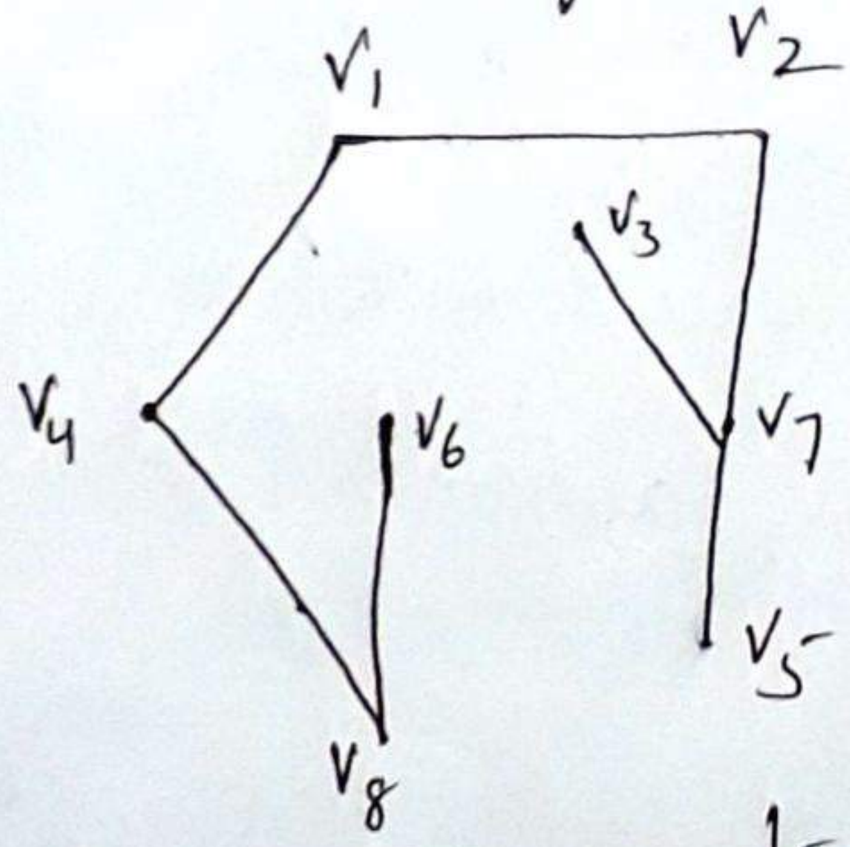
So we go to step 3. Since $v = v_3$ and v_3 is not the first vertex, we go to step 4.

Now, we back track from v_3 and assign the vertex v_7 to v, and then we return to step 2.

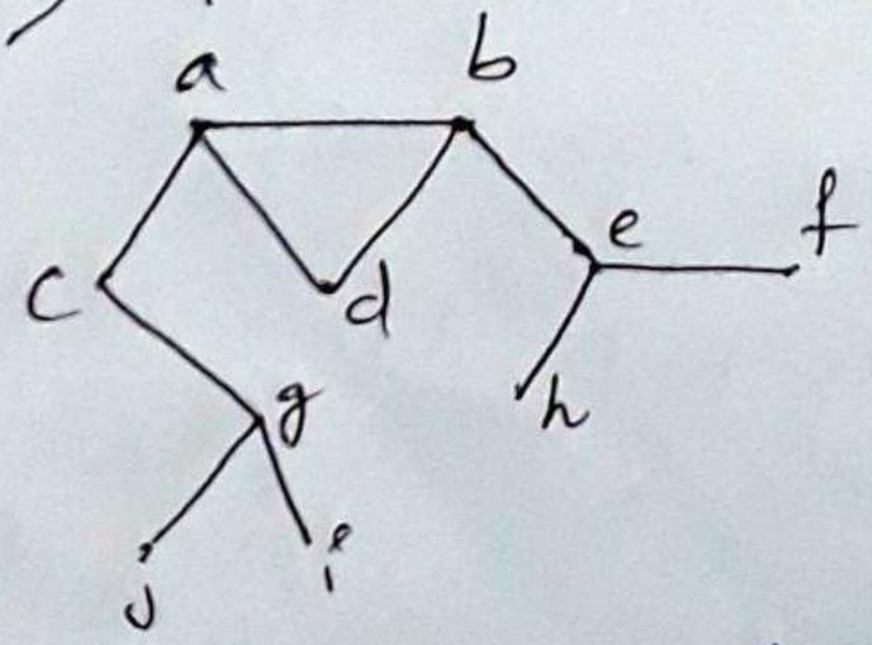
At this time we see that the edge $\{v_7, v_5\}$ can be added to T.

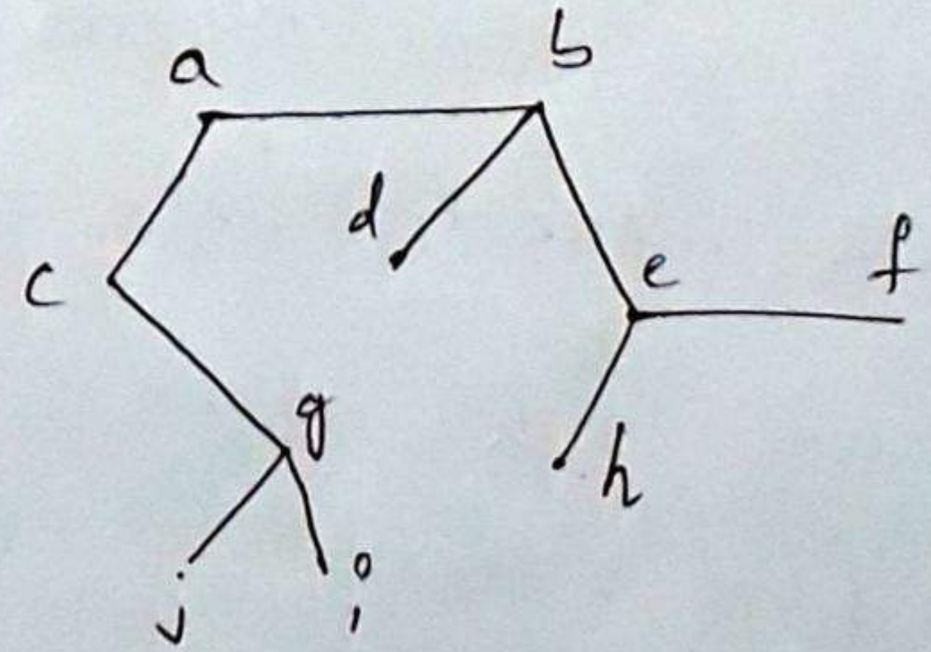
continuing the process, we attach the edges $\{v_1, v_4\}$, $\{v_4, v_8\}$ and finally $\{v_8, v_6\}$ (in this order) to T.

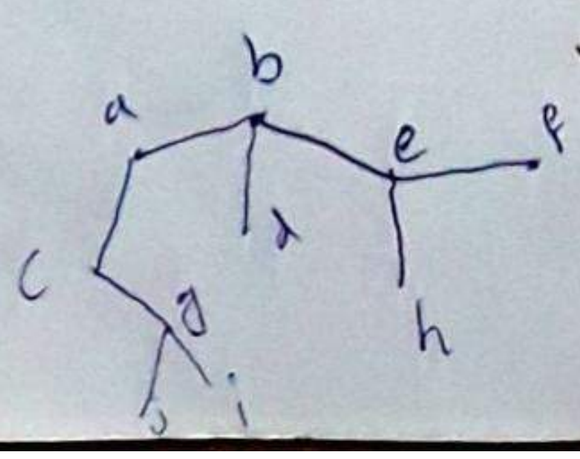
Thus, a spanning tree that consists of the edges $\{v_1, v_2\}$, $\{v_2, v_7\}$, $\{v_7, v_3\}$, $\{v_7, v_5\}$, $\{v_1, v_4\}$, $\{v_4, v_8\}$, $\{v_8, v_6\}$ has been determined. This depth-first search spanning tree is shown in figure.



2) Find spanning tree by using DFS







BFS Algorithm :

Let $G = (V, E)$ be a connected graph of order n, with vertices labeled $v_1, v_2, \ldots v_n$ in some specified order.

The BFS algorithm specifies the following steps for determining a spanning tree T of G, where in 'v' is a variable to which a vertex of G is assigned.

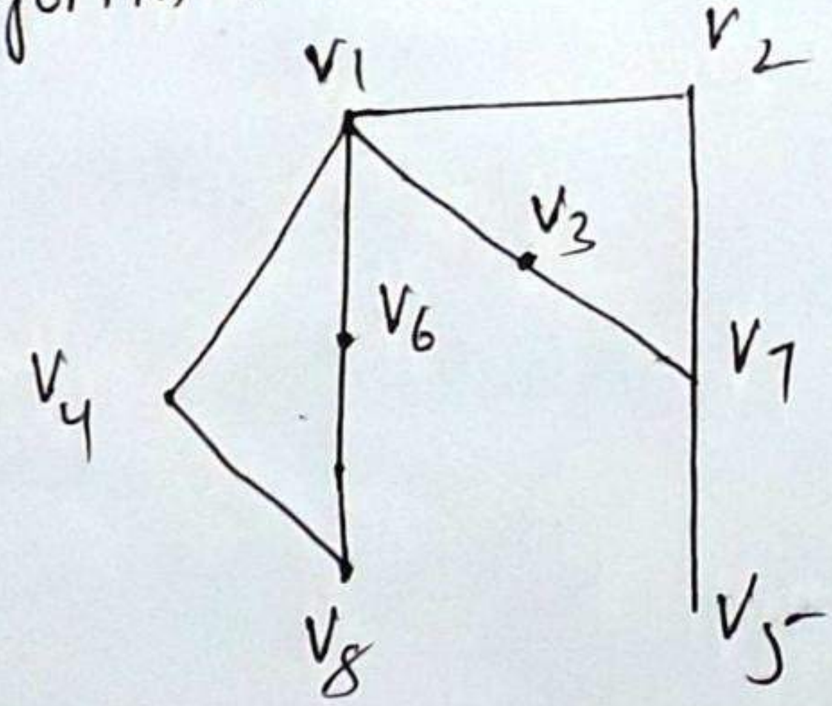
Step 1: Assign the first vertex v_1 and insert this vertex in the queue Q and initialize T as the tree made up of this one vertex v_1. (This vertex will turn out to be the root of the spanning tree T that develops).

Step 2: Delete v from the front of Q. When v is deleted, consider v_K, for each $2 \le K \le n$. If the edge $\{v, v_K\} \in E$ and v_K has not been visited previously, attach this edge to T. If we examine all of the vertices previously visited and obtain no new edge, the tree T (generated to this point) is the desired spanning tree.

Step 3: Insert the vertices adjacent to each v. (from step 2) at the rear of the queue Q, according to the order in which they are (first) visited. Then return to step 2.

The spanning tree of G determined by using the above steps is called the breadth-first search spanning tree for G for the given order of vertices of G.

Ex: Determine the spanning tree by using BFS' Algorithm.



Sol: Start with vertex v_1. Insert v_1 in Q and initialize T as this vertex (which will turn out to be the root of the resulting tree T).

In step 2, we delete v_1 from Q and visit the vertices adjacent to it, namely

v_2, v_3, v_4, v_6. (These vertices have not been previously visited). This results in our attaching to T the edges $\{v_1, v_2\}$, $\{v_1, v_3\}$, $\{v_1, v_4\}$, $\{v_1, v_6\}$.

continuing to step 3, we insert v_2, v_3, v_4, v_6 (in the order) in Q and return to step 2.

Now v_2, v_3, v_4, v_6 are deleted from Q, and the vertices adjacent to them (and not previously visited) are visited. This results in the vertices v_7, v_8 and the edges $\{v_2, v_7\}$ and $\{v_4, v_8\}$ which are attached to T.

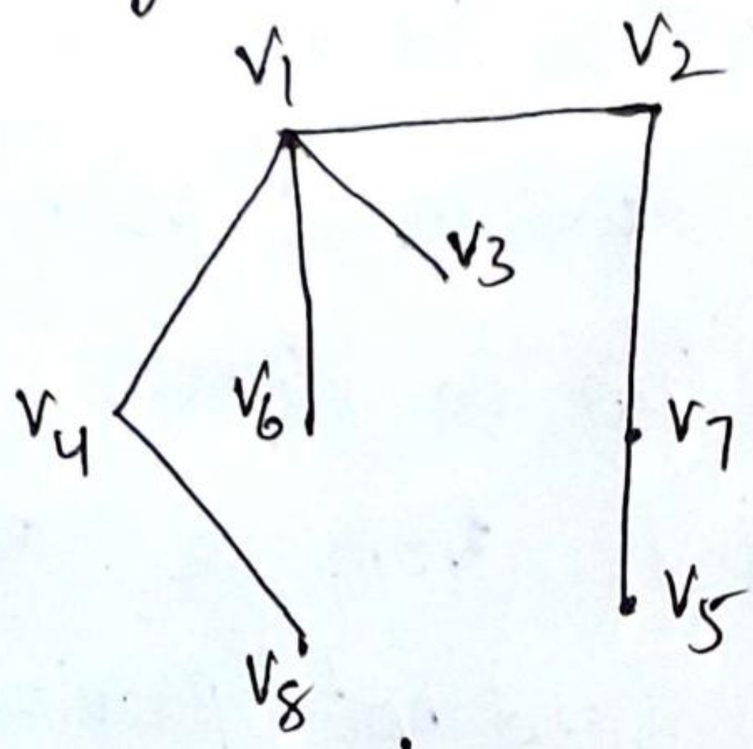
Next, we go to step 3, and insert v_7, v_8 in Q. Returning to step 2, we delete each of these vertices from Q and visit the vertex v_5 which is adjacent to v_7 and which is previously unvisited, and attach the edge $\{v_7, v_5\}$ to T. The vertex v_8 has no previously unvisited adjacent vertex.

At this stage, all the vertices of G have been visited, and we stop the process.

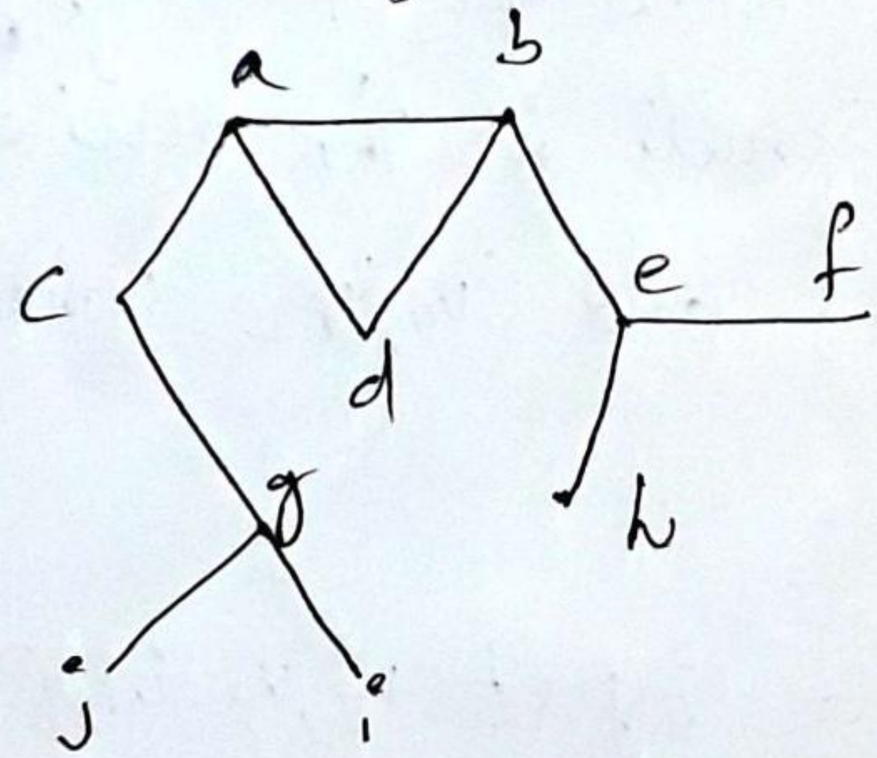
Thus, a spanning tree that consists of the edges $\{v_1, v_2\}$, $\{v_1, v_3\}$, $\{v_1, v_4\}$, $\{v_1, v_6\}$, $\{v_2, v_7\}$, $\{v_4, v_8\}$, $\{v_7, v_5\}$

Scanned by TapScanner

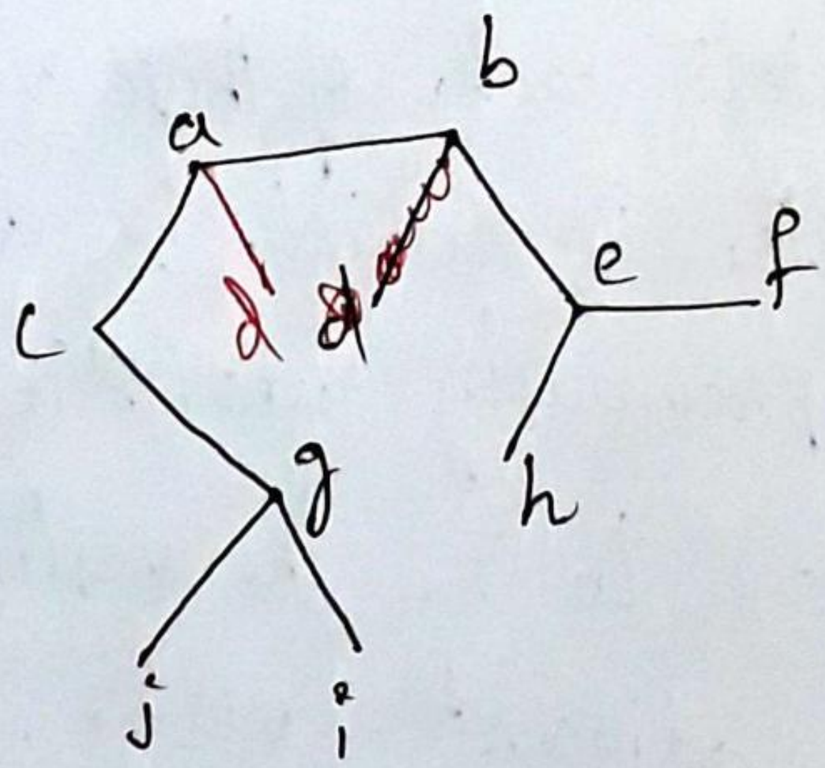
~~has~~ has been determined. This breadth-first search spanning tree is shown in figure.



2)



Sol:



Minimal spanning Tree :

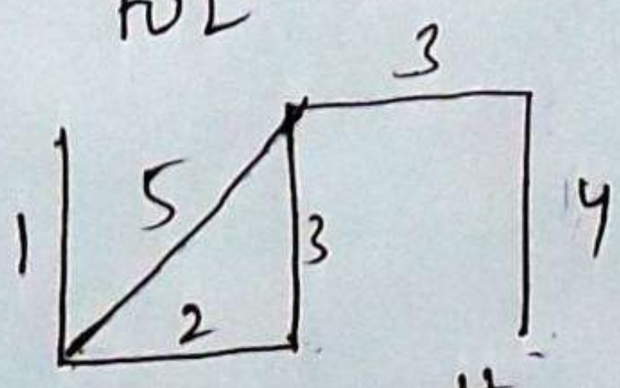
weighted graph: Let G be a graph and suppose there is a positive real number associated with each edge of G. Then G is called a weighted graph and the positive real number associated with an edge e (of G) is called the weight of the edge e.

Minimal spanning Tree:

Let G be a connected weighted graph. A minimal spanning tree of G is a spanning tree of G whose total weight is as small as possible.

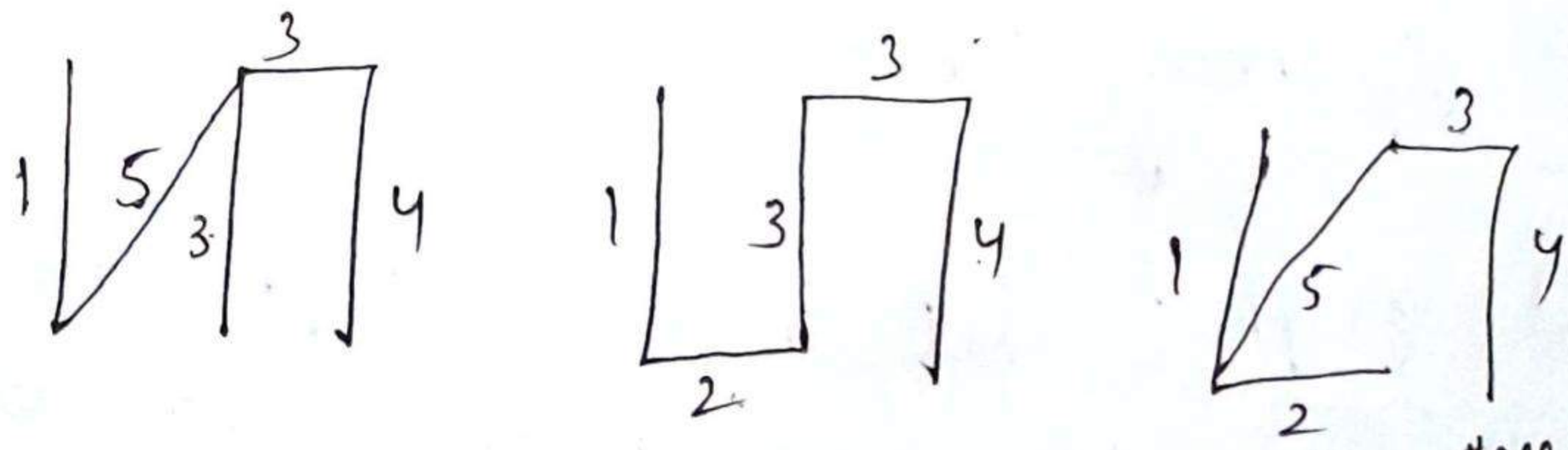
Note: Minimal spanning tree is not unique but their weight must be same.

examples for connected weighted graph



fig(1)

This graph has three spanning trees as shown below:



we observe that, the weights of these three spanning trees are 16, 13, & 15 respectively.

Of these trees, the second one has the minimum weight. This is a minimal spanning tree for the graph fig (4).

Algorithms for Minimal spanning Tree:

There are several methods of constructing minimal spanning trees. The two famous and important algorithms for constructing minimal spanning tree are shown below

(1) Kruskal's algorithm

(2) Prim's Algorithm.

Scanned by TapScanner

1) Kruskal's Algorithm:

The working rule for the Kruskal's method (usually called Kruskal's algorithm) may stated as follows:

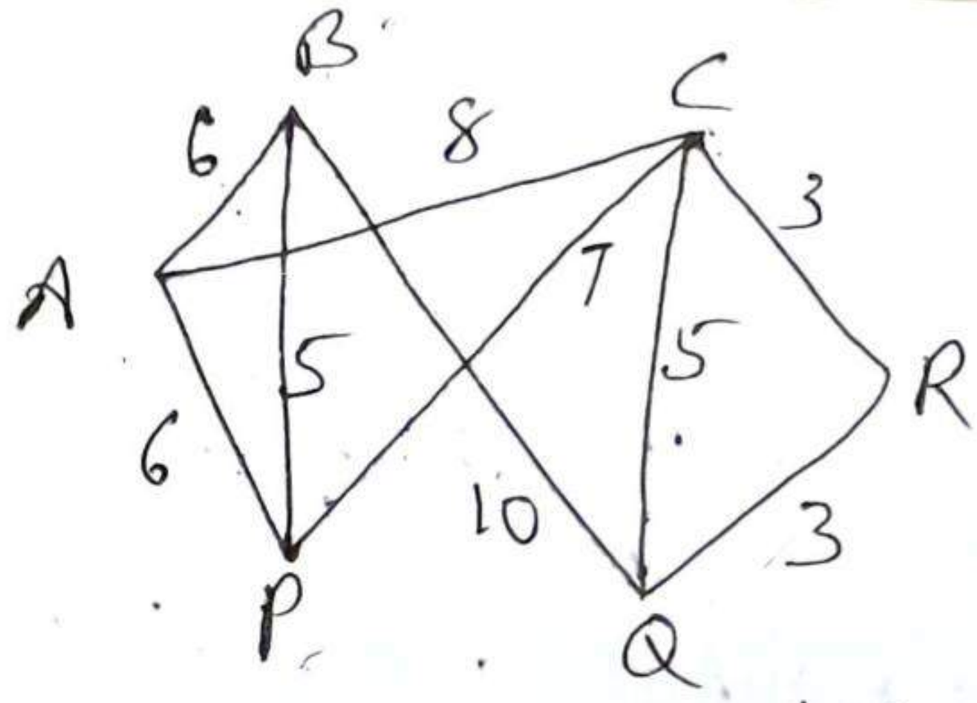
Step1: Given a connected, weighted graph G with 'n' vertices, list the edges of G in the order of increasing weights (non-decreasing).

Step2: Starting with a smallest weighted edge, proceed sequentially by selecting one edge at a time such that no cycle is formed.

Step3: Stop the process of step 2, when (n-1) edges are selected. These (n-1) edges constitute a minimal spanning tree of G.

Ex:1) Using the Kruskal's Algorithm, find a minimal spanning tree of the weighted graph shown below.

Scanned by TapScanner



6 ↗

Sol: we observe that the given graph has 6 vertices. hence a spanning tree thereof will have 5 edges (branches).

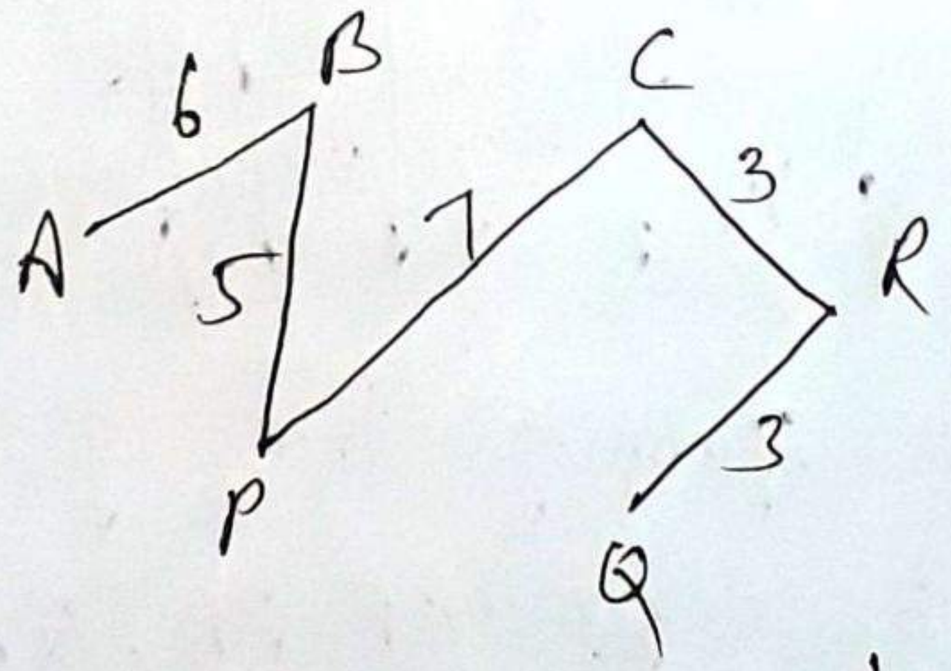
Let us put the edges of the graph in the increasing order of their weights and successively select 5 edges in such a way that no cycle is created. This scheme is summarized in the following Table.

Edge	CR	QR	BP	CQ	AB	AP	PC	AC	BQ
weight	3	3	5	5	6	6	7	8	10
select	yes	yes	yes	No	yes	No	yes		

observe that CQ is not selected because CR and QR have already been selected and the selection of CQ would have created a cycle

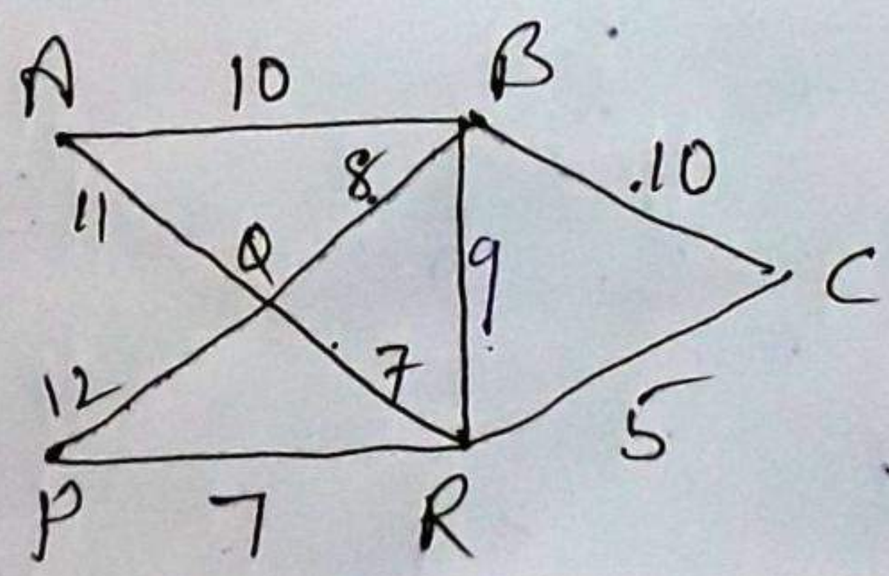
Further, AP is not selected because it would have created a cycle along with BP and AB which have already been selected. We have stopped the process when exactly 5 edges are selected.

Thus, a minimal spanning tree of the given graph contains the five edges CR, QR, BP, AB, CP. This tree is shown in fig.



The weight of this tree is 24 units.

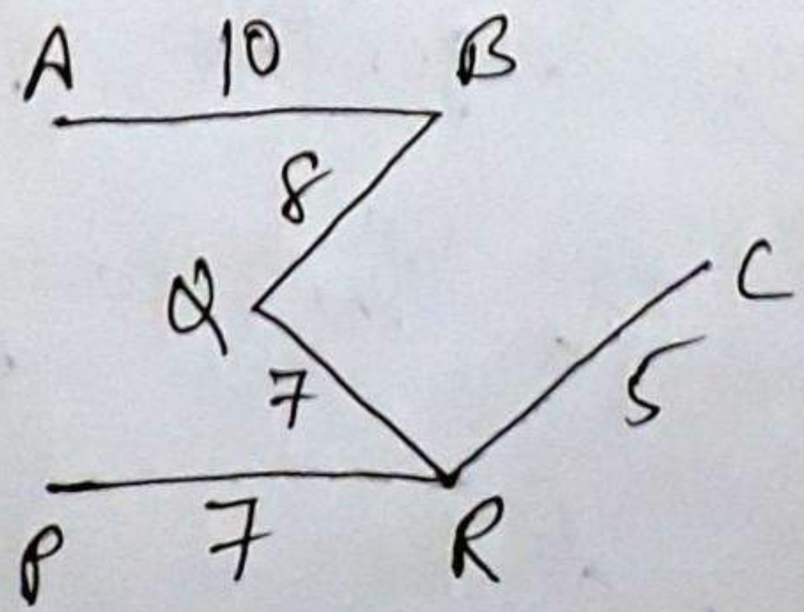
2) Using Kruskal's algorithm, find a minimal spanning tree for the weighted graph shown below:



Sol: The given graph has 6 vertices and therefore a spanning tree thereof will have 5 edges. Let us put the edges of the graph in the non-decreasing order of their weights and go on selecting 5 edges one by one in such a way that no cycle is created. The scheme is summarized in the following table.

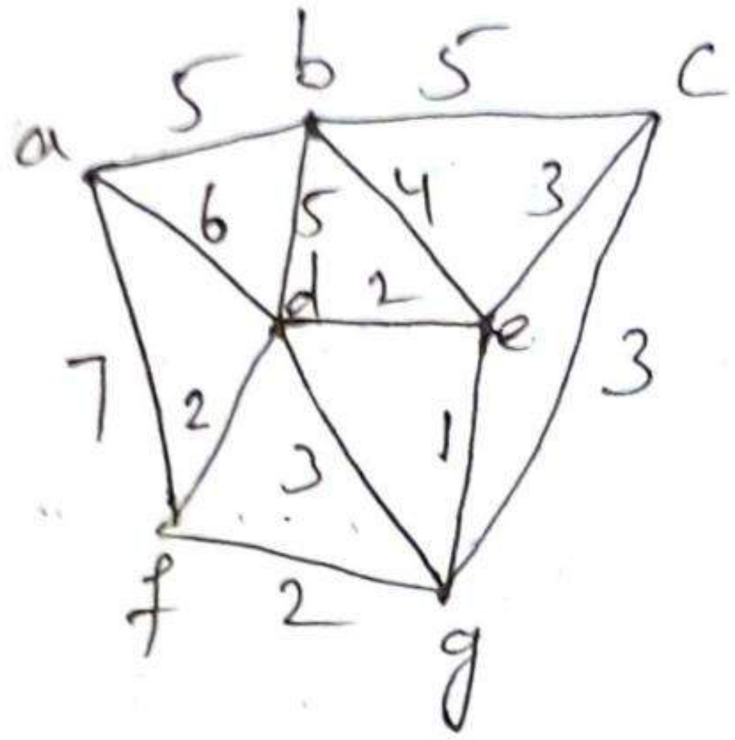
Edge	CR	PR	QR	BQ	BR	AB	BC	AQ	PQ
weight	5	7	7	8	9	10	10	11	12
Select?	Y	Y	Yes	Yes	NO	Yes			

Thus, a minimal spanning tree of the given graph contains 5 edges CR, PR, QR, BQ, AB. The tree is shown in figure.

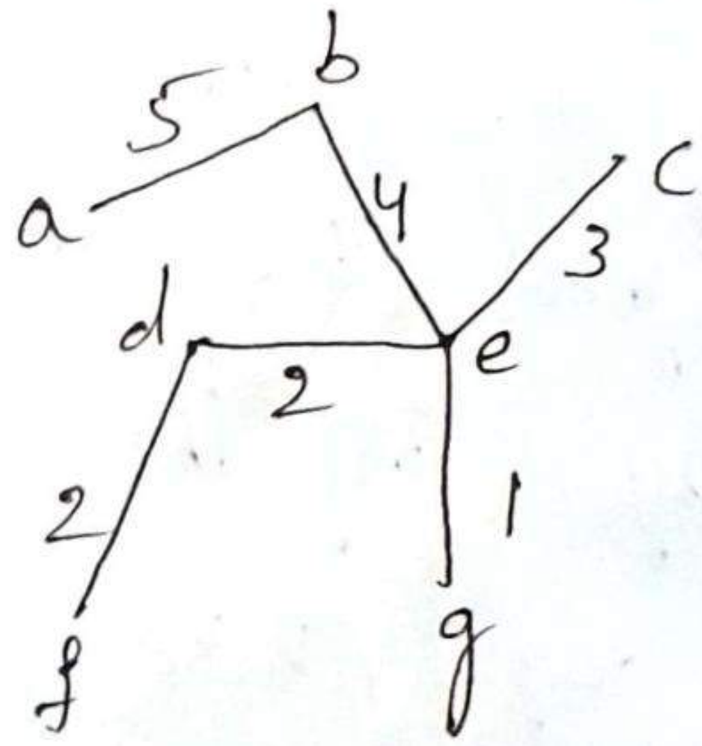


The weight of the tree is 37 units.

3) Apply kruskal's algorithm to the weighted graph (52)



Ans:



2) Prim's Algorithm

The working rule for the Prim's method (usually called prim's Algorithm) may be stated as follows:

Step 1: Given a connected, weighted graph G with n vertices, assign 'n' names (say: $v_1, v_2, \ldots v_n$, or A, B, C and so on) to these vertices, and prepare a $n \times n$ table in which the weights of all edges are shown.

The entries in the table will be symmetric with respect to the diagonal and no entries

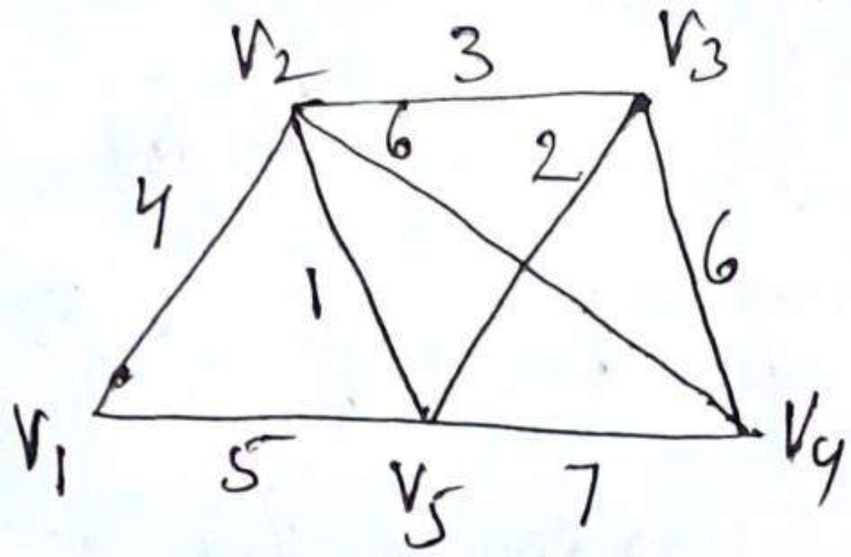
appear on the diagonal. Indicate the weights of the non-existing edges as ∞.

<u>Step 2:</u> Start from the vertex v_1 (or A, as the case may be) and connect it to its nearest neighbor (i.e to the vertex which has the smallest entry) in the v_1-row, say v_k. Now consider the edge $\{v_1, v_k\}$ and connect it to its closest neighbor (i.e to a vertex, other than v_1 and v_k, that has the smallest entry among all entries in v_1- and v_k-rows). Let this vertex be v_m.

<u>Step 3:</u> Start from the vertex v_m and repeat the process of step 2. Stop the process when all the 'n' vertices have been connected by (n-1) edges. These (n-1) edges constitute a minimal spanning tree

Problems

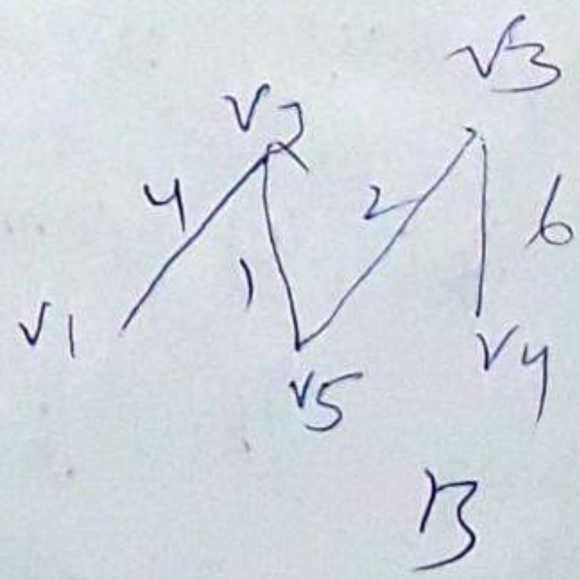
1) Using prim's algorithm, find a minimal spanning tree for the weighted graph shown below:



Sol: we observe that the graph has 5 vertices. Therefore, a minimal spanning tree thereof will have 4 edges.

Let us tabulate the weights of the edges b/w every pair of vertices as shown below:

	v_1	v_2	v_3	v_4	v_5
v_1	–	4	∞	∞	5
v_2	4	–	3	6	1
v_3	∞	3	–	6	2
v_4	∞	6	6	–	7
v_5	5	1	2	7	–



Now let us start with the first row (v_1-row) and pick the smallest entry therein. This is 4

which corresponds to the edge $\{v_1, v_2\}$. By examining all the entries in v_1- and v_2-rows, we find that the vertex other than v_1 and v_2 which corresponds to the smallest entry is v_5 (the smallest entry being 1). Thus, v_5 is closest to the edge $\{v_1, v_2\}$.

Let us connect v_5 to the edge $\{v_1, v_2\}$ at v_2.

Let us now examine the v_5-row and note that the smallest entry is 1 which corresponds to the edge $\{v_5, v_2\}$.

By examining all entries in v_2- and v_5-rows. we find that the vertex other than v_2 and v_5 which corresponds to the smallest entry is v_3.

Thus, v_3 is closest to the edge $\{v_2, v_5\}$.

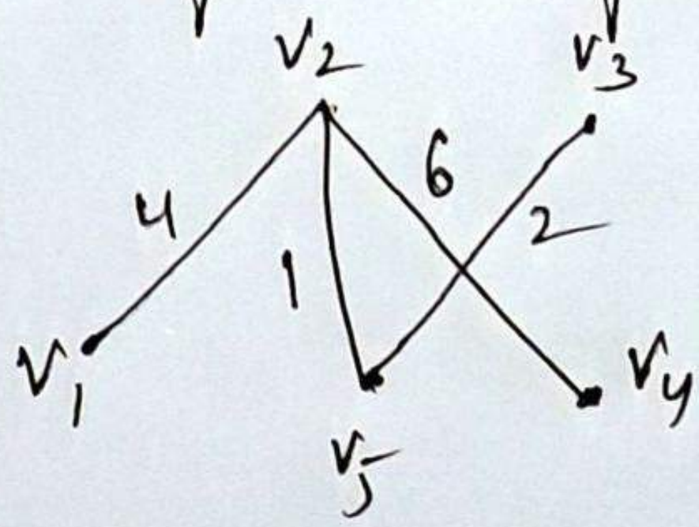
Let us connect v_3 to the edge $\{v_2, v_5\}$ at v_5.

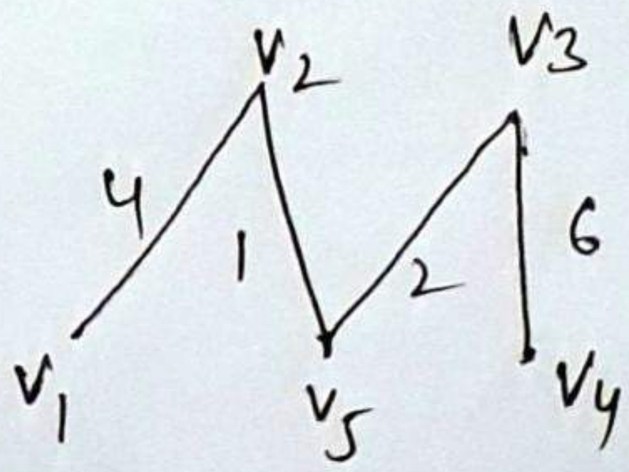
Thus the edges $\{v_1, v_2\}$, $\{v_2, v_5\}$, $\{v_5, v_3\}$ belong to a minimal spanning tree. The vertex left over at this stage is v_4 which is joined to v_2, v_3 and v_5 in the given graph. Among the edges that contain v_4, the edges $\{v_2, v_4\}$ and $\{v_3, v_4\}$

have equal minimum weights. Therefore, we can include either of these edges in the minimal spanning tree.

Accordingly, the edges $\{v_1, v_2\}$, $\{v_2, v_5\}$, $\{v_5, v_3\}$ together with the edge $\{v_2, v_4\}$ or the edge $\{v_3, v_4\}$ constitute a minimal spanning tree.

Thus, for the given graph, there are two minimal spanning trees as shown in figure. ~~The weight of each of these trees is~~





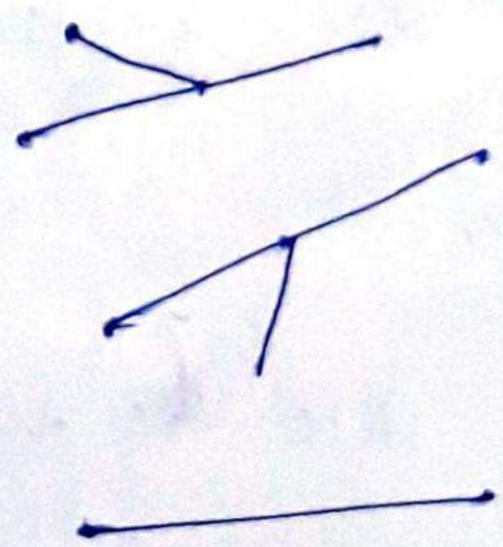
The weight of each of these trees is 13 units.

2)

disconnected shown right side. So, d is cut vertex.

<u>Forest:</u> Forest is a disconnected graph where each component is a tree.

Ex:

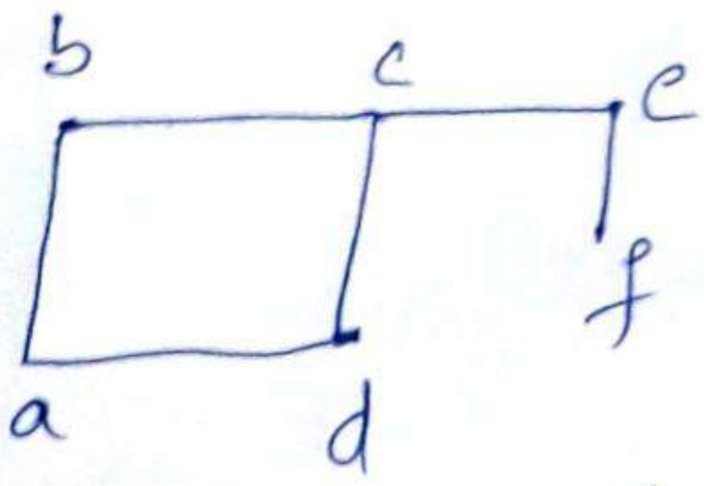


Scanned by TapScanner

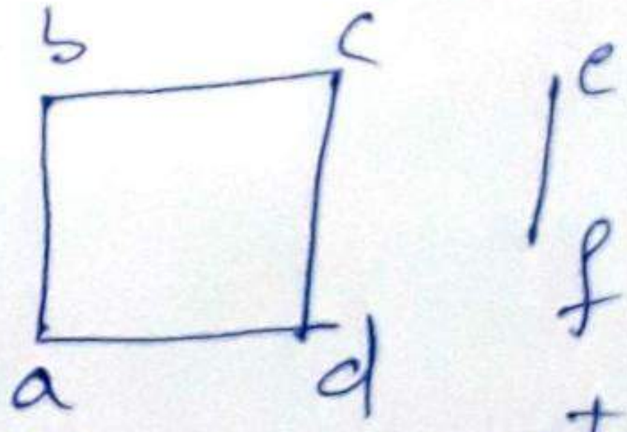
Cut edge (Bridge): If a graph G is connected and E is an edge such that G−E is disconnected. Then E is called the cut edge or Bridge of the graph G.

Note: Every edge of tree is cut edge.

Ex: consider the graph

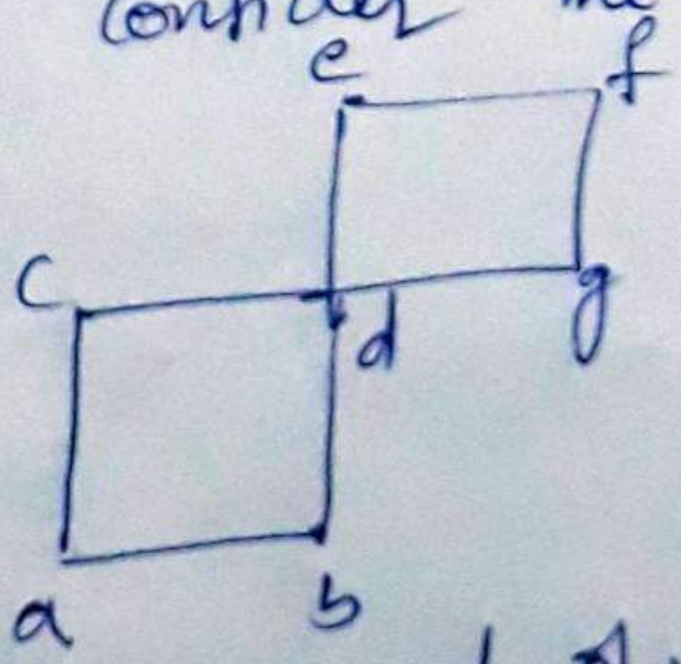


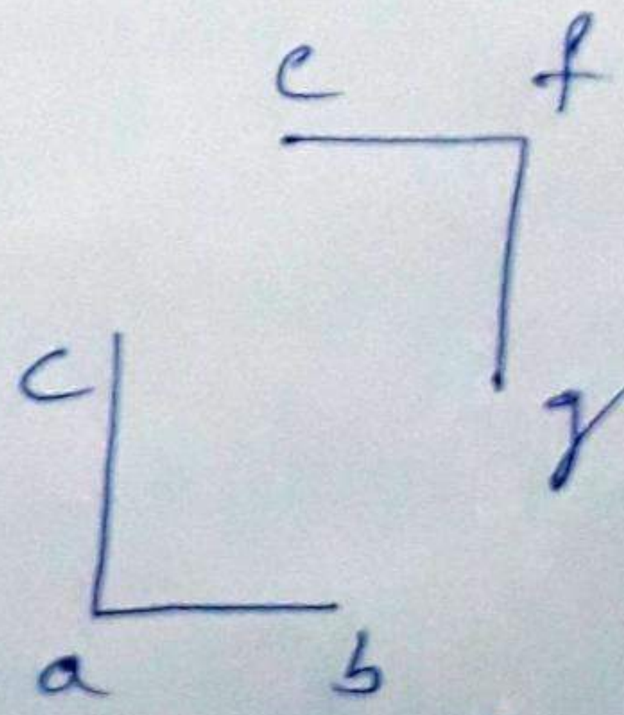
Here the removal of edge {c, e} makes the graph disconnected. This edge is called cut edge.



Cut vertex: If G is a connected graph and v is a vertex such that G−v is disconnected then v is called cut vertex of G.

Ex: consider the graph





Here the removal of vertex 'd' the resulting graph

(continuation is backside of this page)

a ition

Euler Path

An Euler path is a multigraph is a path that includes each edge of the multigraph exactly once and intersects each vertex of the multigraph at least once.

A multigraph is said to be traversable if it has an Euler path.

Euler circuit: An euler circuit is an euler paths whose endpoints are identical.

→ A multigraph is said to be an Eulerian multi-graph if it has an Euler circuit.

Some basic Rules for constructing Hamiltonian Paths and cycles

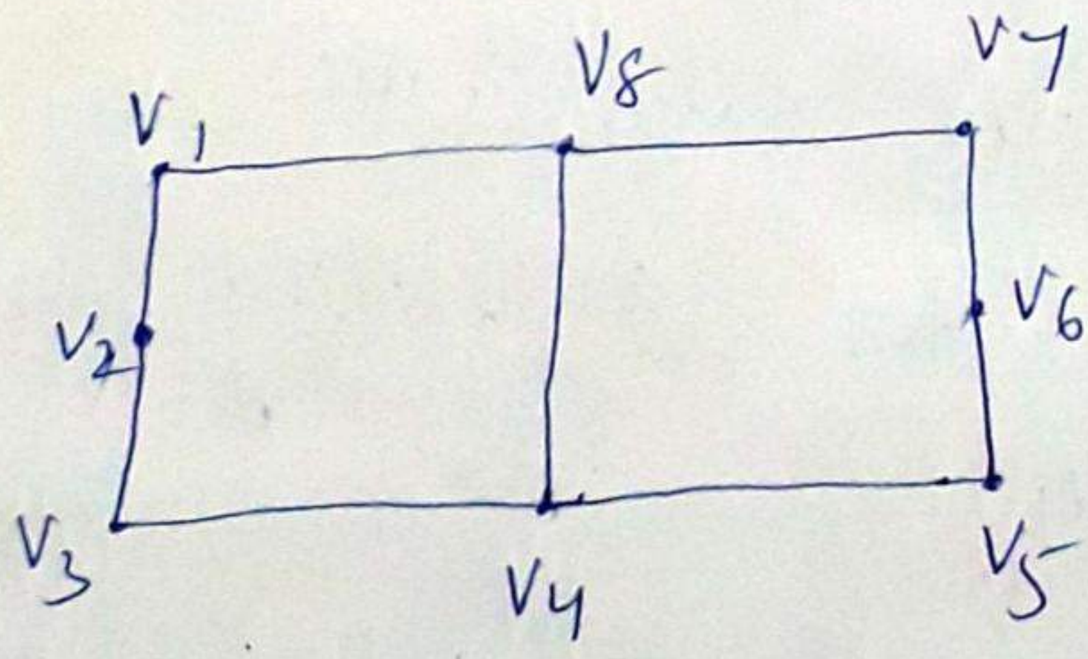
Rule 1: If G has n vertices, then a Hamiltonian path must contain exactly (n-1) edges, and a Hamiltonian cycle must contain exactly 'n' edges.

Rule 2: If a vertex v in G has degree 'k', then a Hamiltonian path must contain at least one

edge incident on 'v' and at most two edges incident on v.

In particular, both edges incident on a vertex of degree two will be contained in every Hamiltonian cycle. In sum: there cannot be three or more edges incident with one vertex in a Hamiltonian cycle.

<u>Rule 3</u>: No cycle that does not contain all the vertices of G can be formed when building a Hamiltonian path or cycle.



<u>Rule 4</u>: once the Hamiltonian cycle we are building has passed through a vertex v, then all other unused edges incident on v can be deleted because only two edges incident on v can be included in a Hamiltonian cycle.