Def 5: If $G = (V, E)$ is a graph and 'V' has finite number of elements then G is said to be a finite graph

Example: G_1, G_2, G_3, G_4 given above are all finite graphs.

Def 6: If $G = (V, E)$ is a finite graph, then $|V(G)|$ denotes the number of vertices in G and is called as **order of G**

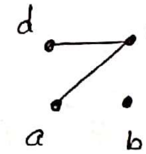
$|E(G)|$ denotes the number of edges in G and is called the **size of G**

Example:

order of G_5 is '4'

size of G_5 is '2'.

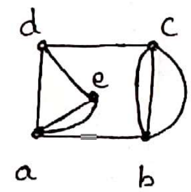
G_5:



Def 7: In a graph $G = (V, E)$ if there are more than one edge between a pair of vertices then 'G' is called a **Multi Graph**.

Example: In G_6,

G_6:



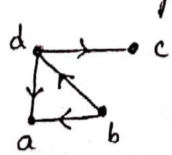
there are more than one edge between the vertices b, c and a, e.

So, G_6 is a multi Graph.

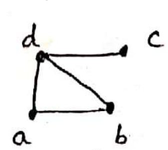
Def 8: A non directed graph 'G' obtained by ignoring the direction of edges of a digraph 'D' is called underlying nondirected graph of D.

Example:

D:



G:

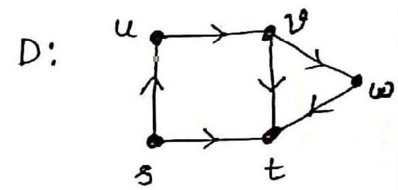


Def 9: In a digraph 'D' an edge (u, v) is said to be 'incident from u' and to be 'incident to v'.

If 'v' is a vertex of 'D' then the number of edges incident to the vertex 'v' is called the in-degree of the vertex 'v' and it is denoted by $deg_D^+(v)$

The number of edges incident from 'v' is called out-degree of 'v' and is denoted by $deg_D^-(v)$

Example:



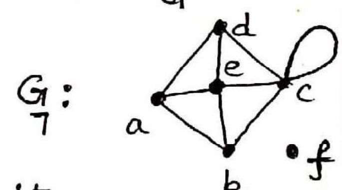
In the digraph D,

$deg^+(v) = 1 \quad deg^-(v) = 2$

$deg^+(s) = 0 \quad deg^-(s) = 2$

Def 10: If $G = (V, E)$ is a nondirected graph, the number of edges passed through a vertex 'v' is called degree of v in G and it is denoted by $deg_G(v)$

Example: $deg(b) = 3$; $deg(e) = 4$



Note: If there is a loop at 'v' then it contributes '2' to the $deg_G(v)$

Example: In G_7 $deg(c) = 5$

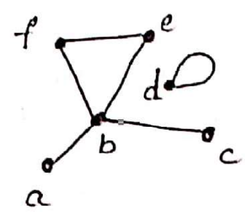
Def 11: In a graph 'G' a vertex of degree zero is called an isolated vertex

Example: In G_7 'f' is an isolated vertex

Def 12 : Let $G = (V, E)$ be a graph / digraph. If 'u' and 'v' are two vertices in G such that they are directly connected by an edge then we say that u and v are adjacent vertices or neighbors

Example :

G_8 :



In the graph G_8

a, b are neighbors (or) adjacent vertices

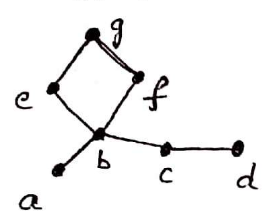
but a and e , b and d are not neighbors.

Notation : Let $G = (V, E)$ be a graph.

$$\delta(G) = \min\{\deg(v) \mid v \in V(G)\}$$

$$\Delta(G) = \max\{\deg(v) \mid v \in V(G)\}$$

Example: Find $\delta(G)$ and $\Delta(G)$ for the following graph G :



$$\delta(G) = \min\{\deg(a), \deg(b), \deg(c), \deg(d), \deg(e), \deg(f), \deg(g)\}$$

$$= \min\{1, 4, 2, 1, 2, 2, 2\}$$

$$= 1$$

$$\Delta(G) = \max\{\deg(a), \deg(b), \deg(c), \deg(d), \deg(e), \deg(f), \deg(g)\}$$

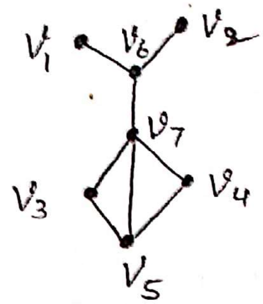
$$= \max\{1, 4, 2, 1, 2, 2, 2\}$$

$$= 4$$

Def 13: Let 'G' be a graph. If the vertices $v_1, v_2, \ldots v_n$ of G are in the increasing order of their degrees then then the sequence $(d_1, d_2, \ldots, d_n)$ where d_i = degree (v_i) is called the degree sequence of G

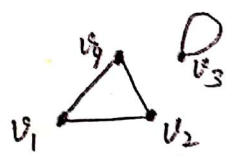
Example: The degree sequence of G_9 is $\langle 1, 1, 2, 2, 3, 3, 4 \rangle$

G_9:

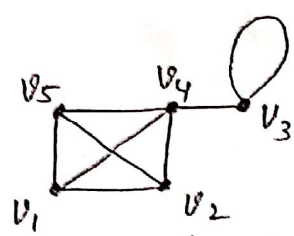


Def 14: If every vertex of a graph 'G' has same degree (say) k. then we say that G is a k-regular graph

Example:



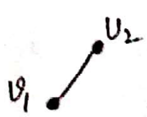
2 – regular graph



not 3 – regular graph

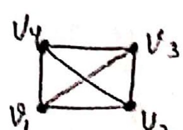
Def 15: A Simple nondirected graph with 'n' mutually adjacent vertices is called a complete graph on n vertices and it is denoted by K_n

Example:

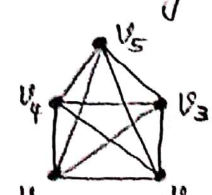


K_2

K_3



K_4



K_5

Note: In K_n (i.e. comple graph on n vertices) degree of every vertex is $(n-1)$

Theorem 1:- The First theorem of Graph Theory (or) The sum of degrees theorem

If $V = \{v_1, v_2, ---, v_n\}$ is the vertex set of a nondirected graph G, then $\sum_{i=1}^{n} deg(v_i) = 2|E|$

If G is a directed graph, then

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

proof:

case i, Let $G = (V, E)$ be a nondirected graph.

and $V = \{v_1, v_2, ---, v_n\}$

If 'e' is any edge of G that connects two vertices (say) v_i, v_j then

'e' contributes '1' to the $deg(v_i)$ and also contributes '1' to the $deg(v_j)$

so, in total 'e' contributes '2' to the degree sum $\sum_{i=1}^{n} deg(v_i)$

$\therefore$ Sum of degrees of all vertices in G is equal to twice the number of edges of G

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

case ii

Suppose G is a directed graph

Let $e = \langle v_i, v_j \rangle$ be an edge of G

Scanned by CamScanner

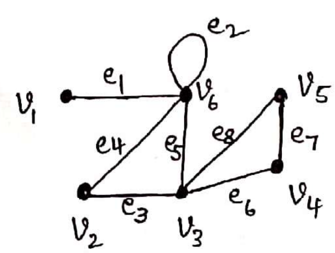
Notation: If 'e' is an edge of a given graph G then $G-e$ denotes the graph obtained by deleting the edge from G.

In general $G-\{e_1, e_2, ---, e_K\}$ denotes the graph obtained by deleting the edges $e_1, e_2, --- e_K$ from G.

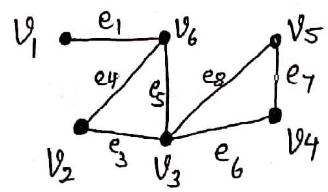
Similarly, $G-v$ denotes the graph obtained by removing the vertex 'v' along with all edges incident on v. In general $G-\{v_1, v_2, ---, v_K\}$ denotes the graph obtained by removing the vertices $v_1, ---, v_K$ along with all edges incident on any of them.

Example:

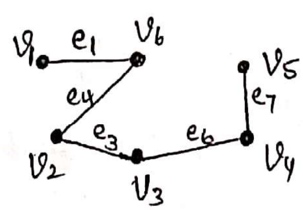
G :



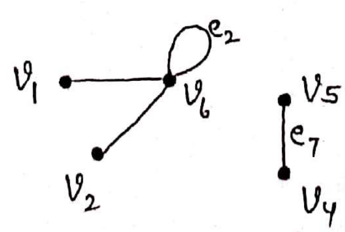
$G-e_2$:



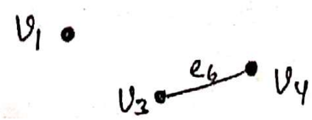
$G-\{e_2, e_5, e_8\}$:



$G-v_3$:



$G-\{v_6, v_5, v_2\}$:



Isomorphism of Graphs

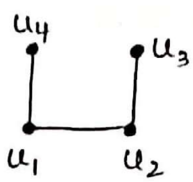
Def : Two graphs G and G' are isomorphic if there is a function $\phi : V(G) \rightarrow V(G')$ from the vertices of G to vertices of G' such that

i) ϕ is one - one

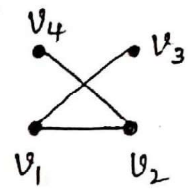
ii) ϕ is on-to

iii) for each pair of $u, v \in V(G)$, u, v are adjacent in G if and only if $\phi(u)$, $\phi(v)$ are adjacent in G'

Example : G :



G' :



Define $\phi : V(G) \longrightarrow V(G')$ by

$\phi(u_1) = v_1$ $\phi(u_2) = v_2$ $\phi(u_3) = v_4$ and $\phi(u_4) = v_3$

Clearly ϕ is a bijection from $V(G)$ to $V(G')$

u_1, u_2 are adjacent in G

$\phi(u_1), \phi(u_2)$ i.e., v_1, v_2 are adjacent in G'

u_2, u_3 are adjacent in G

$\phi(u_2), \phi(u_3)$ i.e., v_2, v_4 are adjacent in G'

u_1, u_4 are adjacent in G

$\phi(u_1), \phi(u_4)$ i.e., v_1, v_3 are adjacent in G'

$\therefore$ $\phi : V(G) \rightarrow V(G')$ is an isomorphism

Hence the graphs G and G' are isomorphic

***
Note: If two graphs G and G' are isomorphic then

i. $|V(G)| = |V(G')|$

ii. $|E(G)| = |E(G')|$

iii. If $v \in V(G)$, then $\deg_G(v) = \deg_{G'}(\phi(v))$, thus the degree sequences of G and G' are the same

iv. If $\{v, v\}$ is a loop in G, then $\{f(v), f(v)\}$ is a loop in G' and in general,

if $v_0 - v_1 - v_2 - \cdots - v_{k-1} - v_k = v_0$ is a cycle of length k in G, then $\phi(v_0) - \phi(v_1) - \phi(v_2) - \cdots \phi(v_{k-1}) - \phi(v_k)$ is a cycle of length 'K' in G'.

If G and G' are isomorphic then all the above four properties holds good

This is equivalent of saying that if any one of the above four properties fails then the graphs G and G' are not isomorphic

It is also important to note that two graphs G and G' are isomorphic if and only if their complement are isomorphic. This statement can be easily concluded from the definitions of Isomorphism and complement of a graph.

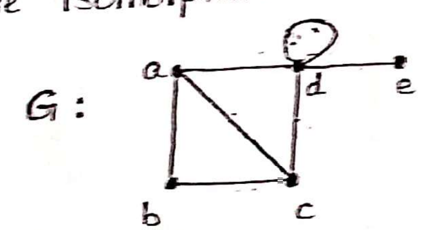
***

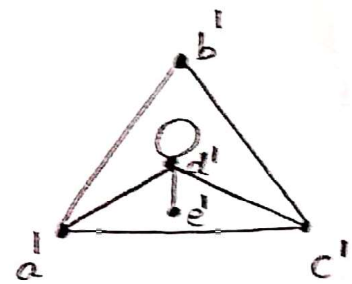
Adjacency matrix: If $v_1, v_2, ---, v_n$ are the vertices of G, then the adjacency matrix for this ordering of the vertices of G is the matrix $A = [a_{ij}]_{n \times n}$ where $a_{ij} = 0$ if the vertices v_i, v_j are not adjacent

$= 1$ if the vertices v_i, v_j are adjacent/neighbours

Note: If G and G' are two graphs and that $\phi : V(G) \to V(G')$ is a bijection. Let A be the adjacency matrix for the vertex ordering $v_1, v_2, ---, v_n$ of G and A' be the adjacency matrix for the vertex ordering $\phi(v_1), \phi(v_2), --- \phi(v_n)$. Then ϕ is isomorphism if and only if $A = A'$

Problem: Show that the following graphs G and G' are isomorphic.





Sol: G and G' are graphs each containing 5 vertices and 7 edges. Their degree sequences $\langle 1, 2, 3, 3, 5 \rangle$ are same. Let us now construct an isomorphism between $V(G)$ and $V(G')$

Since in G and G' e, b, d and e', b', d' are the unique vertices of degree 1, 2, 5 respectively, they must be mapped each other

i.e, $b - b'$, $e - e'$ and $d - d'$

Now the remaining vertices are a, c in G and a', c' in G'. let us map $a - c'$ and $c - a'$

$\therefore$ Define a function $\phi : V(G) \to V(G')$ by

$\phi(a) = c'$ $\phi(b) = b'$ $\phi(c) = a'$ $\phi(d) = d'$ $\phi(e) = e'$

Clearly ϕ is one-one and on-to function,

Also ϕ preserves adjacency.

$\therefore$ ϕ is an isomorphism between $V(G)$ and $V(G')$. Hence G is isomorphic to G'

Verification by adjacency matrix

let 'A' be the adjacency matrix for the vertices of G in the order a, b, c, d, e.

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

let 'A'' be the the adjacency matrix for the vertices of G' in the order $f(a), f(b), f(c), f(d), f(e)$.

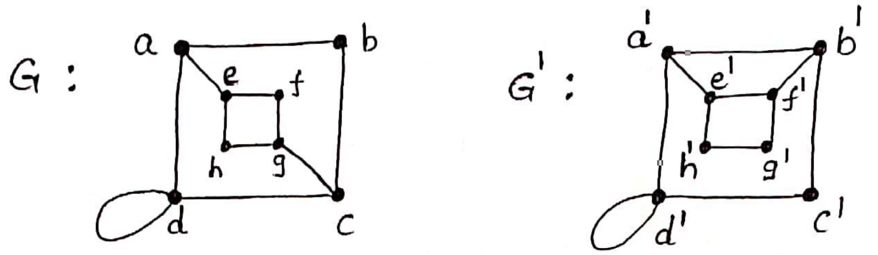
i.e., in the order c', b', a', d', e'

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

Here $A = A'$

$\therefore$ ϕ is an isomorphism

Problem: Show that the two graphs G and G' given below are not isomorphic.



Soln: $|V(G)| = 8 = |V(G')|$

and $|E(G)| = 11 = |E(G')|$

The degree sequence of G and G' are same

i.e., $\langle 2, 2, 2, 3, 3, 3, 4 \rangle$

since G and G' contains only one vertex of degree '4'. these vertices d, d' must be mapped

Now to find an isomorphism, the neighbors of 'd' should be mapped to neighbors of 'd''

In graph G, 'd' has two neighbors namely the vertices 'a' and 'c', where $\deg(a) = \deg(c) = 3$

but 'd'' has two neighbors namely the vertices a' and c', where $\deg(a') = 3$ and $\deg(c') = 2$

i.e., the neighbours of 'd' and 'd'' have different degrees and hence they can't be mapped

$\therefore$ The graphs G and G' are not isomorphic.

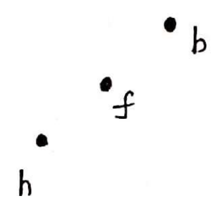
Another method:

If the two graphs G and G' are isomorphic then the respective subgraphs induced by the vertices of same degree would be isomorphic.

Let us consider the subgraphs H and H' induced by the vertices of ~~se~~ degree '2'

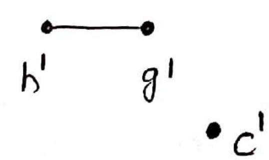
In G, {b, f, h} are the vertices of degree '2'.
The subgraph induced by these vertices (say) H is

H :



In G', {b', g', h'} are the vertices of degree '2'.
The subgraph induced by these vertices (say) H' is

H' :

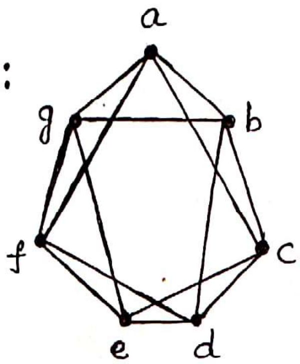


The subgraphs H and H' of G and G' resply., are not isomorphic

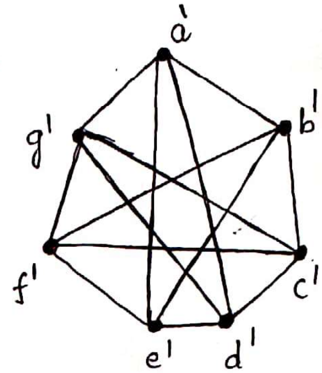
Hence G and G' are not isomorphic.

<u>Problem</u>: The graphs G and G' given below are isomorphic.

G :



G' :



<u>Soln</u>:

<u>Method : 1</u>

Given graphs G and G' both have 7 vertices and 14 edges and every vertex in G and G' has degree 4

Now we try to find an isomorphism between V(G) and V(G')

First let us map a to a'

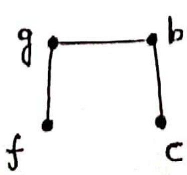
To construct an isomorphism between V(G) and V(G') we have to map adjacent vertices (neighbors) of 'a' to adjacent vertices (neighbors) of a'

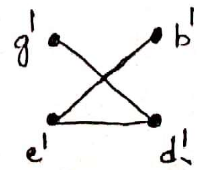
The neighbors of 'a'' are : b', d', e', g'

The neighbors of 'a' are : b, c, f, g

Now we have to map these vertices so that adjacency is preserved.

For this, let us consider the subgraphs of G and G' induced by b, c, f, g and b', d', e', g' respectively.





To preserve the adjacency let us map

$g - e'$, $b - d'$, $c - g'$ and $f - b'$

Now d, e are the remaining unmatched Vertices in G and 'c'' and 'f'' are in G'

suppose 'd' is mapped to f' then in G 'b' and 'd' are adjacent but their images 'd'' and 'f'' not adjacent in G'. so let us map 'd' to 'c'' and 'e' to 'f''

i.e., Define $\phi : V(G) \rightarrow V(G')$ as follows

$\phi(a) = a'$ $\phi(c) = g'$ $\phi(e) = f'$ $\phi(g) = e'$

$\phi(b) = d'$ $\phi(d) = c'$ $\phi(f) = b'$

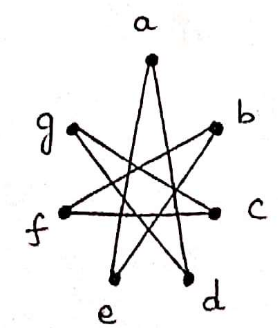
Then ϕ is a bijective function that preserves adjacency.

$\therefore G \cong G'$

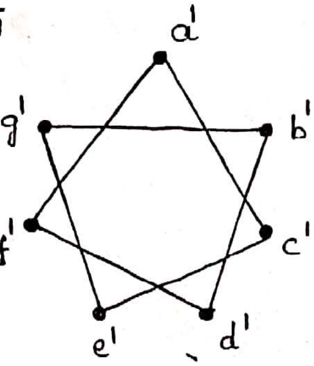
Method - 2: In this method we prove $G \cong G'$ by showing that their complements are isomorphic

Let us draw the complements $\overline{G}$ and $\overline{G'}$ of G and G' respectively.

$\overline{G}$:



$\overline{G'}$



both these complements are just cycles of length '7'

$\bar{G}$: $a - d - g - c - f - b - e - a$

$\bar{G'}$: $a' - c' - e' - g' - b' - d' - f' - a'$

Now consider a map $\phi' : V(\bar{G}) \to V(\bar{G'})$ by

$\phi'(a) = a'$ $\phi'(g) = e'$ $\phi'(f) = b'$ $\phi'(e) = f'$

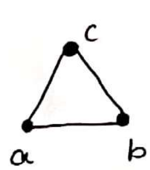
$\phi'(d) = c'$ $\phi'(c) = g'$ $\phi'(b) = d'$

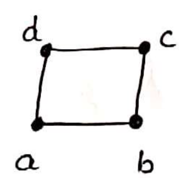
Then ϕ' is a bijective function and preserves adjacency.

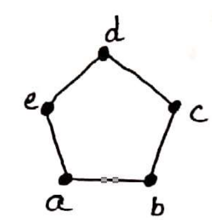
Therefore, $\bar{G}$ and $\bar{G}'$ are isomorphic. Hence the graphs G and G' are isomorphic

Special graphs :

Def : A cycle graph of order 'n' is a connected graph whose edges form a cycle of length 'n'. and it is denoted by C_n

C_3 : 

C_4 : 

C_5 : 

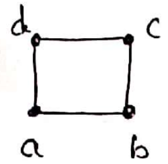
Def : A nondirected graph $G = (V, E)$ is said to be a bipartite graph if 'V' can be partitioned into two sets 'M' and 'N' in such a way that every edge of 'G' joins a vertex in 'M' to a vertex in N.

Further, if every vertex of M is adjacent to every vertex of N then such a bipartite graph is called complete bipartite graph

The complete bipartite graph that is partitioned into sets M and N such that $|M| = m$ and $|N| = n$ is denoted by $K_{m,n}$ (consider m and n are such that $m \le n$)

Example:

G_1 :



For the graph G_1

$M = \{a, c\}$ $N = \{b, d\}$ forms a bipartition.

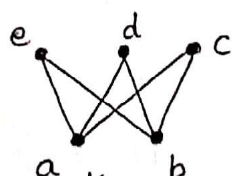
$\therefore$ G_1 is a bipartite graph

Further, every vertex of M is adjacent to every vertex of N

$\therefore$ G_1 is a complete bipartite graph. and $|M| = 2 = |N|$.

So, G_1 can be denoted as $K_{2,2}$

Example:

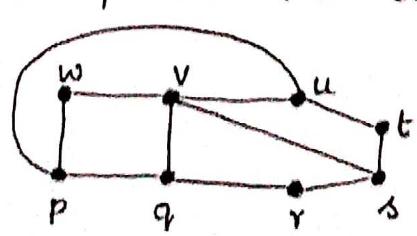


$K_{2,3}$

Here $M = \{a, b\}$ $N = \{c, d, e\}$

Exercise: Find the following graph is bipartite (or) complete bipartite (or) a $K_{m,n}$ graph.

G :

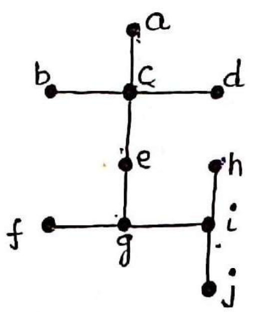


Trees and their Properties

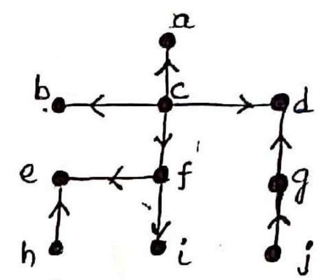
Def 1: A simple graph $G = (V, E)$ is said to be a **tree** if there exists a unique simple nondirected path between each pair of vertices of G.

Example:

G_1:



G_2:



Note: A tree may be a digraph or a nondirected graph.

Def 2: In a tree $G = (V, E)$ if a vertex (say) 'v' is designated as a root then G is called **rooted tree**.

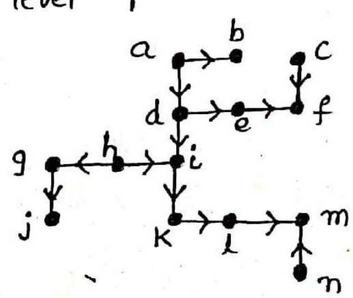
Note: In a tree any vertex may be designated as a root.

Def 3: The **level** of a vertex 'v' in rooted tree $G = (V, E)$ is the length of the (non-directed) path to 'v' from the root.

Example: In G_1 given above if the vertex 'c' is designated as root then vertex 'h' is at level 4. and vertex 'b' is at level '1'.

Exercise:

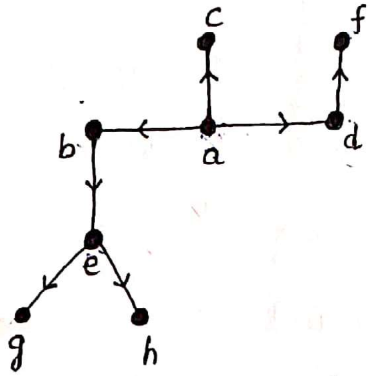
G':



In the graph G' what are the levels of the vertices g, c, l if 'm' is the root?

Def 4: In a rooted tree $G=(V,E)$ if there is a root 'v' such that each vertex in 'G' can be reached from 'v' by a directed path then G is called a directed tree.

Example: G_3:



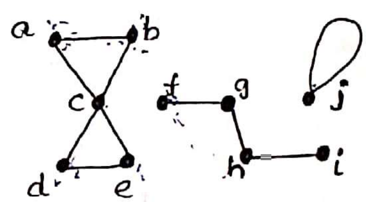
In G_3 if the vertex 'a' is designated as a root then there is a directed path from 'a' to each vertex of G.

Def 5: In a graph $G=(V,E)$ two vertices a, b are said to be connected if and only if there is a nondirected path from 'a' to 'b'

Example:

In the graph G_4, the vertices a, e are connected but c, h are not connected.

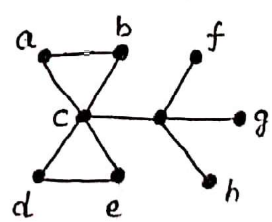
G_4:



Def 6: A graph $G=(V,E)$ is connected if and only if each pair of its vertices is connected

Example: G_5 is a connected graph but G_4 in the above example is not connected.

G_5:



Def 7: A connected subgraph H of a graph is said to be a component of 'G', if for each connected subgraph 'F' of G where $H \subseteq F \subsetneq G$, $V(H) \subseteq V(F)$ and $E(H) \subseteq E(F)$. then $H=F$

Example: For the graph G_4 there are 3 components.

Note: i) In otherwords, A component 'H' of a graph 'G' is a maximal connected subgraph of 'G' i.e., there is no connected subgraph 'F' of G such that $H \subsetneq F \subsetneq G$

ii) A graph G is connected if and only if there is only one component for G that is G itself.

Theorem 1: A simple nondirected graph 'G' is a tree if and only if G is connected and contains no cycles.

Proof: Let $G = (V, E)$ be a simple nondirected graph.

($\Rightarrow$) Suppose that G is a tree.

By definition of tree, there is a unique, simple nondirected path between each pair of vertices of 'G'. Hence 'G' is connected.

If 'G' contains a cycle then any pair of vertices (say u, v) on this cycle are joined by atleast two simple paths. But this is contradiction to the definition of tree. Therefore 'G' contains no cycles.

($\Leftarrow$) conversely, suppose that 'G' is connected and contains no cycles.

Let 'a' and 'b' be any pair of vertices of 'G'.

Since G is connected, there is atleast one simple path that joins 'a' and 'b'.

Now we shall prove that such a simple path from 'a' to 'b' is unique.

If possible suppose that, there are two different simple paths P_1 and P_2 from 'a' to 'b'

Since P_1 and P_2 are different paths, there must be a vertex v_1 (say) on both paths such that the vertex following v_1 on P_1 is different from the vertex following v_1 on P_2

Since P_1 and P_2 terminates at 'b', there there is a first vertex after v_1 say v_2 which is common for both of the paths P_1 and P_2

Thus the part of P_1 from v_1 to v_2 together with that part of P_2 from v_1 to v_2 form a cycle in G.

But this is a contradiction because 'G' has no cycles.

∴ Between each pair of vertices there is only one (unique) simple path in 'G'.

Def: A tree 'T' with only one vertex is called a trivial tree. otherwise 'T' is a nontrivial tree

Theorem: 2 In every nontrivial finite tree there is atleast one vertex of degree '1'.

Proof: Let $G = (V, E)$ be a nontrivial finite tree

Start at any vertex V_1 of G. If $\deg(V_1) = 1$ then we are through

Suppose $\deg(V_1) \neq 1$ then travel along any edge that incident at 'V_1' to reach another vertex (say) V_2. If $\deg(V_2) = 1$ we are through

otherwise (i.e $\deg(V_2) \neq 1$), go to another vertex V_3 ($V_3 \neq V_1$) along a different edge.

continue this process that produce a path $V_1 - V_2 - V_3 - V_4 - \cdots$ and none of the V_i's is repeated. in this path. Because, if any V_i is repeated then we get a cycle which is not possible in a tree G.

Since the number of vertices in the graph G is finite this path must end somewhere say at V_K. Then $\deg(V_K) = 1$, Since we can enter this vertex but cannot leave it.

Hence the proof of the theorem.

Theorem 3: A tree with 'n' vertices has exactly $(n-1)$ edges.

Proof: Let $G=(V,E)$ be a tree with 'n' vertices.

To prove the theorem we shall use mathematical induction on the number of vertices (n) of 'G'.

If $n=1$, then there are no edges in 'G'. Hence the result is trivial for $n=1$

Assume that, for $k \geq 1$ all trees with 'k' vertices have exactly $(k-1)$ edges.

Let 'G' be a tree with $(k+1)$ vertices.

By a known theorem (theorem 2), we have there is a vertex (say) 'v' in G with $\deg(v)=1$ i.e., there is only one edge (say) 'e' incident at 'v'

Let $G' = G - v$ (i.e., G' is the graph obtained by removing 'v' and the edge 'e' from 'G'.)

then G' is also a tree with k vertices.

By induction hypothesis G' has exactly $(k-1)$ edges.

Now by adding the vertex 'v' and the edge 'e' to G' we get 'G' and then 'G' has exactly $(k-1)+1 = k$ edges.

$\therefore$ The theorem is true for any tree with 'n' vertices where $n \in \mathbb{Z}^+$

Corollary 1: If 'G' is a nontrivial tree then 'G' contains atleast two vertices of degree '1'.

proof: Let $G = (V, E)$ be a nontrivial tree with 'n' vertices.

By first theorem of graph theory,

$$\sum_{v_i \in V(G)} \deg(v_i) = 2|E|$$

But by a known result (theorem 3) we have that G shall have exactly $(n-1)$ edges.

$$\therefore \sum_{v_i \in V(G)} \deg(v_i) = 2(n-1)$$
$$= 2n-2 \quad \text{——— (1)}$$

In a tree G, we know that (by theorem 2) there is atleast one vertex (say) v_1 in G such that $\deg(v_1) = 1$.

suppose 'v_1' is the only vertex with $\deg(v_1) = 1$

$\therefore \deg(v_i) \geq 2, \ \forall \ i = 2, 3, \dots n$

consider $\sum_{i=1}^{n} \deg(v_i) = \deg(v_1) + \sum_{i=2}^{n} \deg(v_i)$

$$\geq 1 + 2\sum_{i=2}^{n} 1 = 1 + 2(n-1) = (2n-1)$$

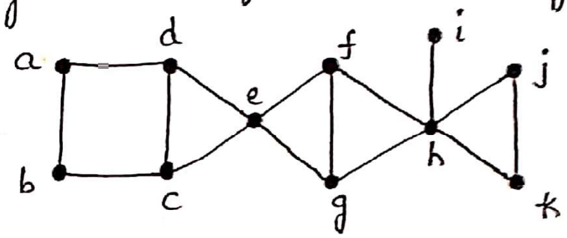
$$\therefore \sum_{i=1}^{n} \deg(v_i) \geq (2n-1) \text{ (or) } (2n-2) \geq (2n-1)$$

which is a contradiction.

$\therefore$ G contains more than one vertex (-atleast two vertices) of degree 1.

3. (update V) Replace 'V' by all the children 'v' in 'T' of the vertices 'x' of 'V' where the edges $\{x, v\}$ were added instep '2'. Go back and repeat step 2 for the new set V.

<u>Example problem</u>: Find a Spanning tree for the graph 'G' using Breadth-first search algorithm.

G: 

<u>Solution</u>: Consider the ordering of the vertices 'abcdefghijk'

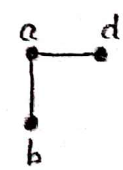
Now we wish to find a spanning tree (say) 'T' for the graph G

Select 'a' as the first vertex for the spanning tree 'T' and designate it as the root of 'T'.

set $V = \{a\}$ T: a •

Now we select all those edges $\{a, x\}$, where 'x' runs from b, c, - - - k in that order that do not form a cycle in 'T'.

$\therefore$ $\{a, b\}$, $\{a, d\}$ are the edges that can be added to 'T' (Now these two edges are called tree edges)

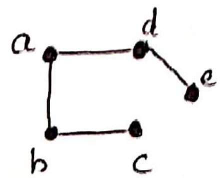
T: 

set $V = \{b, d\}$

Both the vertices 'b' and 'd' in V are at level '1' from the root 'a'. Let us first examine the edges incident at 'b'.

At 'b', include the edge {b,c} as a tree edge. Then at 'd', reject the edge {d,c} since its inclusion would produce a cycle in 'T'. But we can include the edge {d,e}

T:



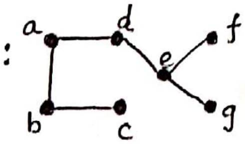
Now set V = {c, e}

At 'c', we reject the edge {c, e}

At 'e', we include the edges {e,f} and {e,g}

set V = {f, g}

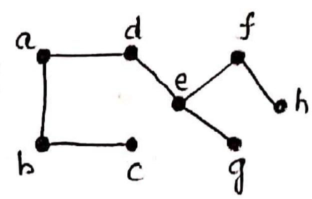
T:



At 'f' we reject the edge {f, g} but include the edge {f, h} for 'T'

At 'g' we can't include the edge {g, h} as it creates a cycle in 'T'

T:

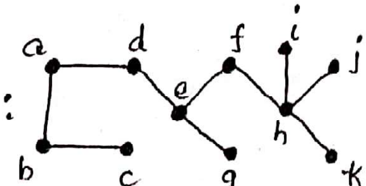


Now set V = {h}

At 'h', we include the edges {h,i}, {h,j}, {h,k} and set V = {i, j, k}

but no edges can be added to 'T' from the vertices i, j, k. So stop here. The tree 'T' obtained above is a spanning tree for the given graph G.

T:



Depth - First Search Algorithm for a Spanning tree

Input : A connected graph G with vertices labelled $v_1, v_2, \ldots v_n$

output : A Spanning tree 'T' for G

Method :

1. (Visit a vertex) Let v_1 be the root of T and set $L = v_1$ (The name 'L' stands for the vertex last visited)

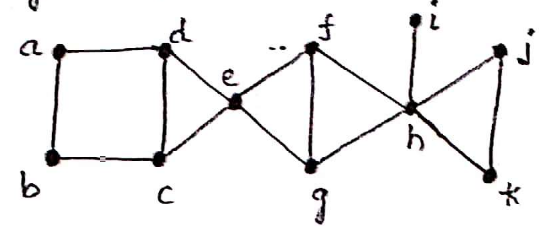
2. (Find an unexamined edge and an unvisited vertex adjacent to 'L')

   For all vertices adjacent to L, choose the edge $\{L, v_k\}$ where 'k' is the minimum index such that adding $\{L, v_k\}$ to 'T' doesnot create a cycle. If no such edge exists, go to step 3; otherwise, add edge $\{L, v_k\}$ to 'T' and set $L = v_k$. Repeat step 2 at the new value for L

3. (Back track or terminate)

   If 'x' is the parent of L in 'T', set $L = x$ and apply step 2 at the new value of L. (coming back to parent of 'L' is known as backtracking
   If on the otherhand, L has no parent in 'T' (i.e., $L = v_1$ (the root)) then the depth - first Search terminates and 'T' is a spanning tree for 'G'

Example problem: Find a spanning tree for the graph 'G' using Depth-first search algorithm.

G :


Solution: consider the ordering of the vertices 'a, b, c, d, e, f, g, h, i, j, k'.

Now we wish to find a spanning tree 'T' for the graph 'G'.

Select 'a' as the first vertex for the spanning tree 'T' and designate it as the root of 'T' (The vertex 'a' is said to be visited)

set $L = a$

T:


Now we find an edge $\{a, x\}$ such that,

i) 'x' is the first label in the designated order of vertices

ii) $\{a, x\}$ is an unexamined edge and 'x' is unvisited vertex.

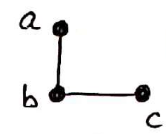
iii) addition of $\{a, x\}$ doesnot create a cycle in 'T'

In the given graph 'G' we select the edge $\{a, b\}$ as a tree edge (Here 'b' is child of 'a' and 'a' is parent of 'b'.)

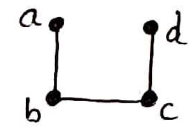
T:


set $L = b$ and find an edge $\{b, x\}$ such that it satisfies all the properties i ii iii with 'a' replaced by 'b'.

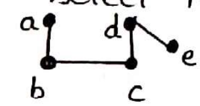
Thus we select the edge {b, c} as a tree edge and set L = c

T:


Now we continue our search at 'c' and select {c, d} and set L = d

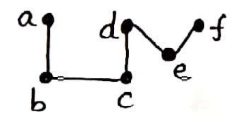
T:


At 'd', first we reject the edge {d, a} as 'a' is already visited then select the edge {d, e} and set L = e

T:


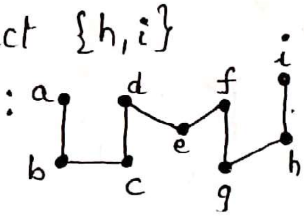
At e, we reject the edge {e, c} and select {e, f}

Continuing in this manner,

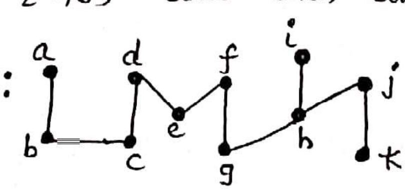
T:


we select {f, g}, reject {g, e}
select {g, h}, reject {h, f} and select {h, i}

At 'i' no edges exists. So, we backtrack to parent of 'i' (i.e 'h') and set L = h

T:


At 'h', select the edge {h, j} and then at 'j' select the edge {j, k}

T:


Now set L = k but reject the edge {k, h}. Since there are no unexamined edges at 'k' backtrack to 'j' and then to 'h' e.t.c....

Eventually we must backtrack all the way back to the root 'a'

∴ The depth-first search terminates here and the tree 'T' obtained above is a spanning tree of 'G'.

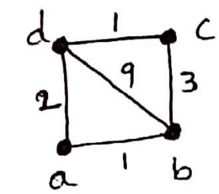
Def: Suppose 'G' is a connected graph and a nonnegative number $c(e)$ (called cost / weight / length) is assigned to each edge 'e' of 'G'. A Subgraph 'H' of the graph G is said to be a minimal spanning tree of G if,

(a) H is a tree

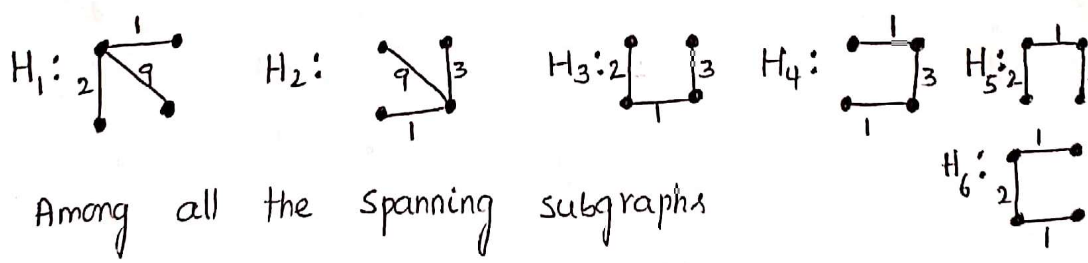
(b) H contains all the vertices of G

(c) $c(H) = \sum_{e \in E(H)} c(e)$ is minimal. (i.e., if H' is and other spanning tree of G then $c(H) \leq c(H')$)

Example: consider the graph G :



consider the spanning trees of G



Among all the spanning subgraphs

$$c(H_6) = 2+1+1 = 4 < c(H_i), \quad i = 1, 2, 3, 4, 5$$

$\therefore$ H_6 is the minimal spanning tree of 'G'

<u>Kruskal's Algorithm for Finding a Minimal Spanning Tree</u>

<u>Input</u>: A connected graph 'G' with nonnegative values assigned to each edge.

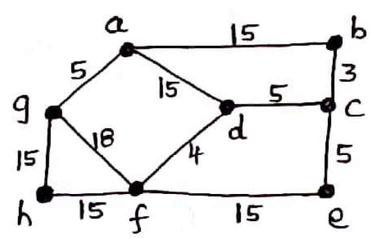
<u>output</u>: A minimal spanning tree 'T' for 'G'.

Scanned by CamScanner

Method :

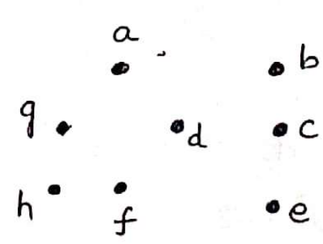
1. Consider the nullgraph with all vertices of 'G'.
2. Select any edge of minimal value that is not a loop. This is the first edge of T. (If there is more than one edge of minimal value, arbitrarily choose one of these edges.)
3. select any remaining edge of 'G' having minimal value that does not form a circuit with the edges already included in 'T'.
4. continue step 3 until 'T' contains (n-1) edges, where 'n' is the number of vertices of 'G'.

Example: Determine a railway network of minimal cost for the cities in the graph G :



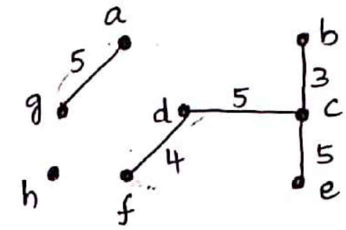
Solution:

Let us consider the null graph formed by all the vertices of G



Select the edges {b, c}, {d, f}, {a, g}, {c, d}, {c, e}. Then the tree 'T' is

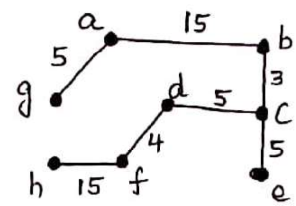
T:



Now we have option to choose only one of {a,b} and {a,d} as selection of both creates a circuit. Let us choose {a, b}

Similarly, we may choose only one of {g,h} and {f,h}. Suppose we choose {f,h} then the total number of edges in 'T' becomes 'T' = (8-1) Hence we stop. The minimal spanning tree of 'G' is

T :



The minimal cost for construction of this tree is

$$3+4+5+5+5+15+15 = 52.$$

Prim's Algorithm for a Minimal Spanning Tree

Input: A connected graph 'G' with nonnegative values (costs) assigned to each edge

output: A minimal spanning tree ('T' say) for 'G'.

Method:

1. (Start) take any vertex v_1 of 'G' as the initial vertex of 'T' and form the set $V=\{v_1\}$
2. (Add new edges) Choose an edge 'e' of 'G' such that,

i. The edge 'e' is not in 'T'

ii. 'e' is incident at any of the vertices of V.

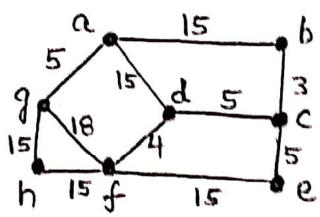
iii. The edge 'e' donot form a circuit when added to 'T'.

iv. 'e' is the edge with minimal cost that has all the properties of i, ii and iii

and add it (the edge 'e') to 'T'. If the number of edges in 'T' reached to $|V(G)|-1$ then stop and in this case 'T' is the minimal spanning tree. otherwise, go to step '3'.

3) (update V): Include the end vertices of all edges that were newly added in the previous step and repeat step '2' for the newset V.

Example: Determine a minimal spanning tree for the graph G:



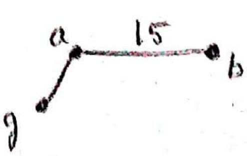
Solution: Let us start with the vertex 'a'

set $V = \{a\}$ T: •a

Now we add the edge $\{a, g\}$ to 'T' and set $V = \{a, g\}$

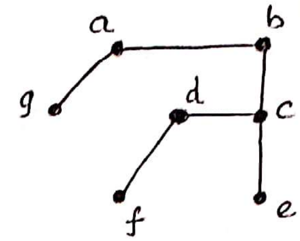
Among the edges incident from the vertices of 'V' the edge $\{a, b\}$ has minimum cost so add it to 'T'

Now $V = \{a, b, g\}$ T:



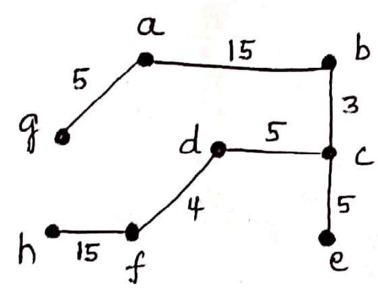
Select the edges $\{b,c\}$, $\{c,e\}$, $\{c,d\}$ and $\{d,f\}$

then T' becomes :



Now $V = \{a,b,c,d,e,f,g\}$

Among the edges incident from the vertices of V we can choose either $\{h,f\}$ or $\{g,h\}$ but not both. Let us choose the edge $\{f,h\}$ then we shall have $7 = (8-1)$ edges in T' so, stop here. The minimum spanning tree T' is



The minimal cost for construction of this tree is.

$$3+4+5+5+5+15+15 = 52.$$

Exercise: Find a minimal spanning tree for each of the graphs by using Kruskal's and prims algorithms. what do you observe?