Tree –

Theorem → If G be an acyclic graph (graph without cycle) with n vertex & K connected components, the G has (n−K) edges.

Solution → Let G be an acyclic graph with n vertices and K connected component $G_1, G_2 \ldots G_K$.
Since all components are seperately. & let each G_i component contain n_i vertices without loop (circuit) then each G_i is a tree with n_i vertices & $n_i - 1$ edges,

So total vertices, $n = n_1 + n_2 + n_3 + n_4 + \ldots + n_K$

So, the total number of edges in G.

No. of edges in G_1 + No. of edges in G_2 + ... + No. of edges in G_K

$= (n_1 - 1) + (n_2 - 1) + (n_3 - 1) + \ldots + (n_K - 1)$

$= (n_1 + n_2 + n_3 + \ldots + n_K) - (1 + 1 + 1 + \ldots + K \text{ times})$

$= n - K$

Theorem → If any tree (with two or more vertices) then there are at least two pendent vertex (vertex of degree one).

proof → Let T be a tree with n vertices $(a_1, a_2 \ldots a_n)$ & $(n-1)$ edges $(e_1, e_2, \ldots e_{n-1})$. Since all the edges are connected with the 2 vertex at a time.
Hence the sum of the degree of all the vertices in, $T = 2 \times (\text{No. of edges}) = 2(n-1)$.
Now, we have to prove that in a tree T there are atleast two vertex of degree one & rest of degree two or higher. ~~Since no vertex~~ in T has (

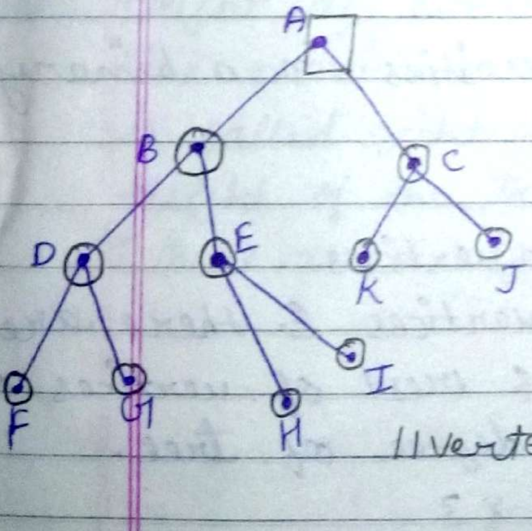
Teacher's Signature:

Scanned by CamScanner

degree so let we assume that there is only one vertex v_1 of degree one & rest (n-1) vertex are of degree two or higher. Then the sum of the degree of vertex are $1 + 2(n-1) = 2n-1$ which is contraduction of $(2n-2)$. Hence there is another vertex of degree one.
Hence if we take 2 vertex of degree of 1 and (n-2) vertex of degree 2 or more then sum of the degree of vertex $= 2 + 2(n-2) = 2n-2$ which is correct => there exist atleast 2 vertices of degree one (Two pendent) Hence proved.

one vertex of degree 2

* Binary tree →
A tree having only one vertex of degree two and rest vertex of degree one or three is called Binary tree.



11 vertex, 5 level

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

proof → Let T(V, E) be a binary tree with n vertices. Since in a binary tree there is only one vertex of degree two & remaining of degree one & three or more.

Let there is K pendent vertex (degree one) and one vertex of degree two and remaining vertex of degree 3. So the total number of degree of graph be

$$K.1 + 2.1 + (n-K-1) \times 3 = K + 2 + 3n - 3K - 3 \quad \text{—— (A)}$$

but we know that each edge associate with two vertex

So sum of the degree $= 2E$ —— (B)

In a tree the no. of vertex = no. of edges+1

$$n = E + 1$$

$E = n - 1$ Sum of the degree $= 2(n-1)$ —— (C)

from (A) & (C)

Total number of degree $= 3n - 2K - 1 \Rightarrow 2n - 2$

$$\Rightarrow n = 2K + 1$$

which is always odd for any integer K.

Hence the no. of vertices in a binary tree is always odd.

Theorem → The number of pendent verities in a binary tree T with n vertices is $\frac{n+1}{2}$

Proof → Let T(V, E) be a binary tree n vertices

Let there are P pendent vertices & there are only one vertex of degree two & rest of vertices of degree 3. Then the total no. degree of tree.

$$\Rightarrow p \times 1 + 2 \times 1 + (n-p-1) \times 3$$

$$\Rightarrow p + 2 + 3n - 3p - 3$$

$$\Rightarrow 3n - 2p - 1 \quad \text{—— (A)}$$

Relation between vertex and edge $\Rightarrow V = E + 1 = E\ (n-1)$ ($V = n$ vertices)

We know the sum of degree $= 2E = 2(n-1)$ —— (B)

from (A) & (B) $\Rightarrow 3n - 2P - 1 = 2n - 2$

Teacher's Signature

Scanned by CamScanner

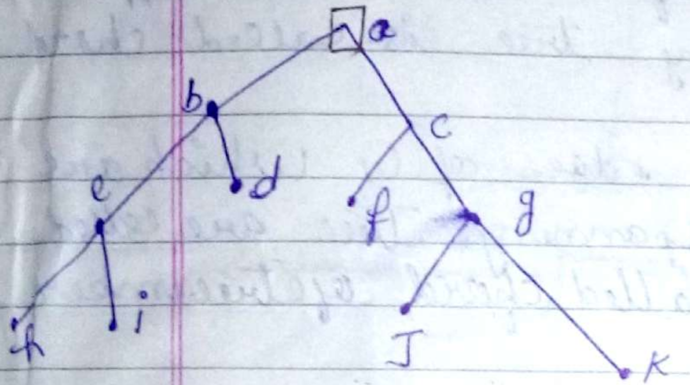
Date: / /

$\Rightarrow n+1 = 2p \Rightarrow p = \frac{n+1}{2}$

* Binary rooted tree →

A Rooted tree is called binary rooted tree if for each vertex v; out degree (v) = 0, 1 or 2 not more than 2, every vertex has at most two children.

Level of vertex → The no. of edges between root and vertex

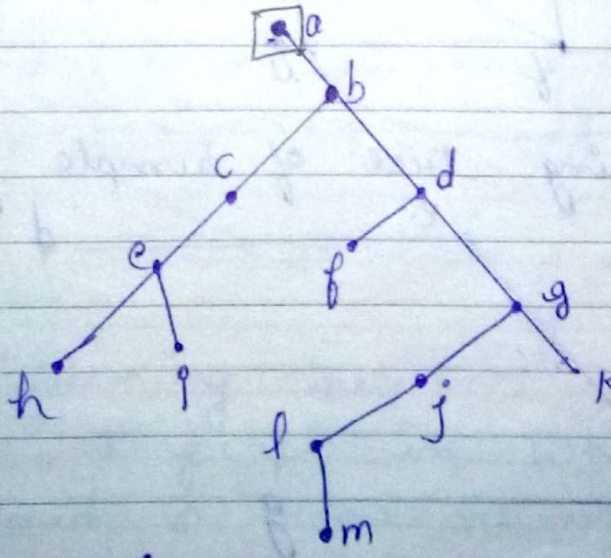


Tree → G_1

* Height of Binary tree →

The maximum level of vertex in a binary tree is called right if tree. The height of a tree G_1 is 3

Height of the tree G_2 is 6 as the level of m is 6 [there are 6 edges between root a and m]



Tree → G_2

Scanned by CamScanner

Date: / /

✱ Spanning tree →

- Let G be a graph with n vertices and e edges then Spanning tree, T of G is a subgraph of G having all vertices of G, it is connected to subgraph of G and it does not have any loop or circuit.

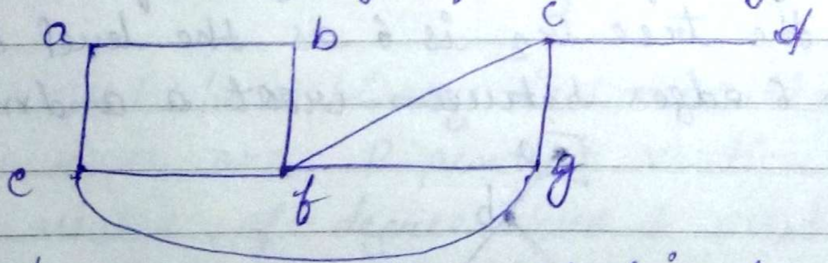
Branch of T → The edges of G that includ in Spanning tree T is called branch of T.

Chord of T → The edges of G that are dropped in Spanning tree is called chord of T.

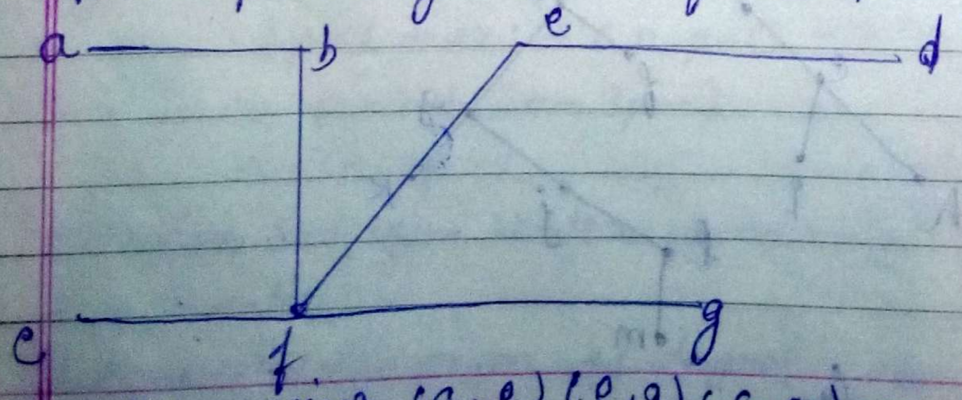
Rank → The number of edges of G which are include in spanning tree are called branchs of T or rank of G their number
$\Rightarrow n-1 = r$.

Nullity → The number of edges of G which are exclude or dropped in spanning tree T are called chord of tree T or nullity of G. nullity $\Rightarrow G = e-(n-1)$

xample →



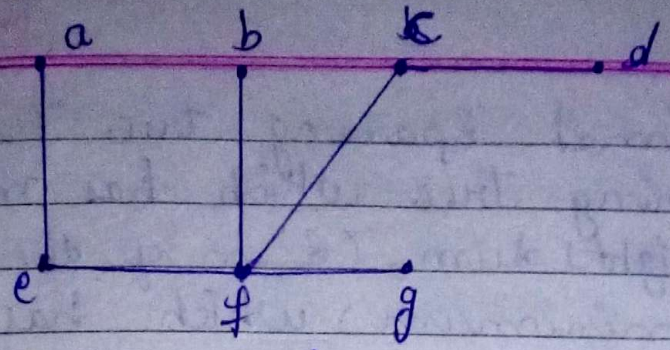
find spanning tree of simple graph.



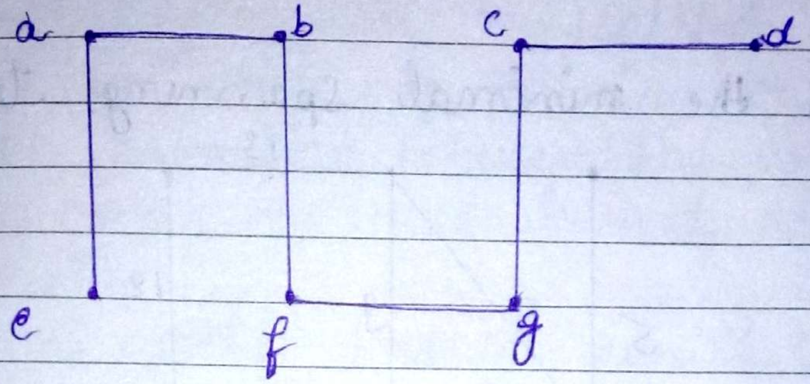
remaining (a, e) (e, g) (c, g)

Teacher's Signature :

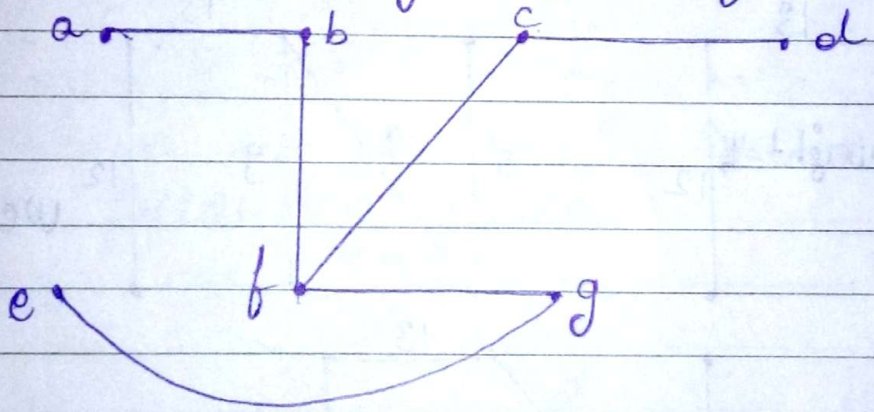
Scanned by CamScanner

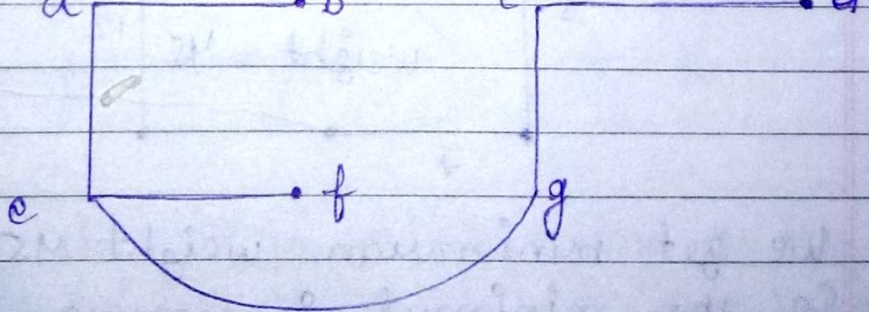


removing (a,b) (e, g) (c, g)



removing (e, f) (e, g) (f, c)



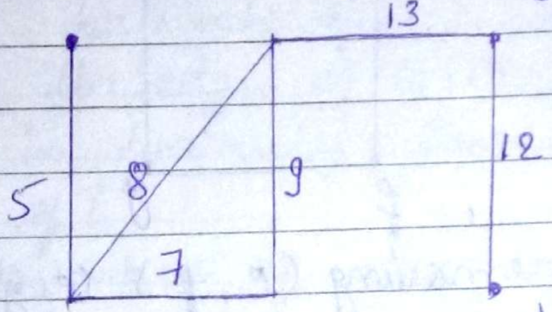


✱ Minimal Spanning tree →
Let G be a graph in which each edge are assigned a real number (weight). Then

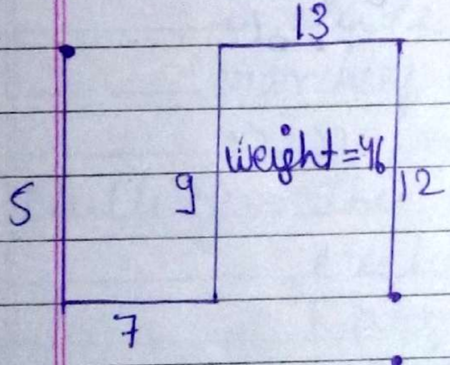
Scanned by CamScanner

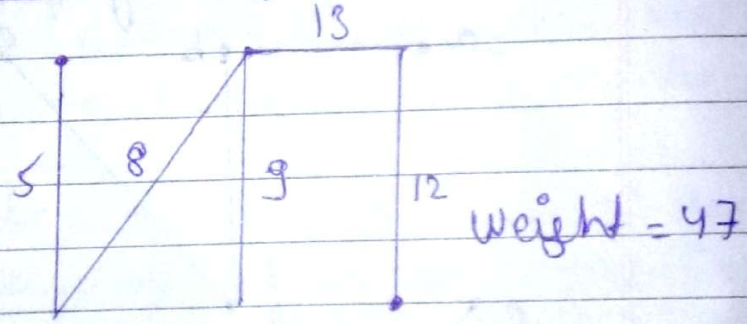
The minimal Spaning tree T of G is that Spaning tree which has minimum length (weight) sum, (Sum of the length of edges is minimum) which has minimum length (weight) Sum. (Sum of the length of edges is minimum)

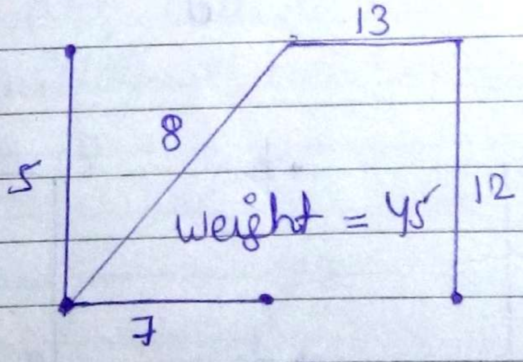
Example → find the minimal spanning tree



Ans –







We get minimum weight 45 for tree T_3.
So the minimal Spanning tree is T_3.

* Method for finding Spanning Tree →
There are so many method for finding Spanning tree some of there as follows –

Scanned by CamScanner

Krushkal Algorithm →

Let G be a graph with n-vertices and e weighted edges then we find the Spanning tree T in following step.

(a) first arrange the edges of G in increasing weights.

(b) Then display the n-isolated vertices.

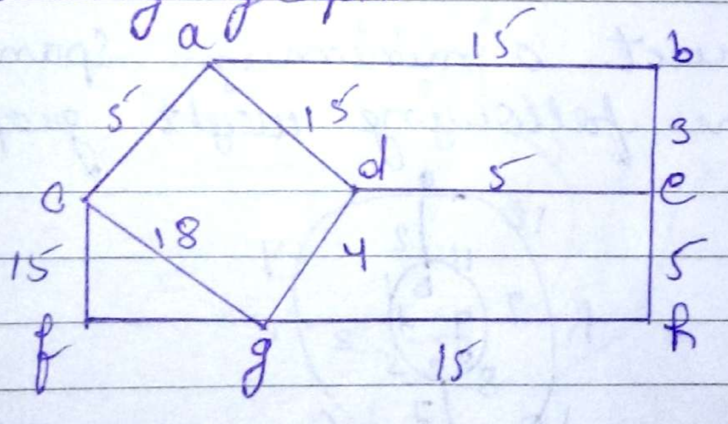
(c) Select the edge of minimum weight and make connection between the vertices.

(d) Then choose the next lowest edges. Make connection.

(e) Continue this process until (n-1) edges cover n vertices without making circuit.

Q.1 Example →

obtain the minimal spanning tree in the following graph.



Step1 → arrange the vertices in ascending order

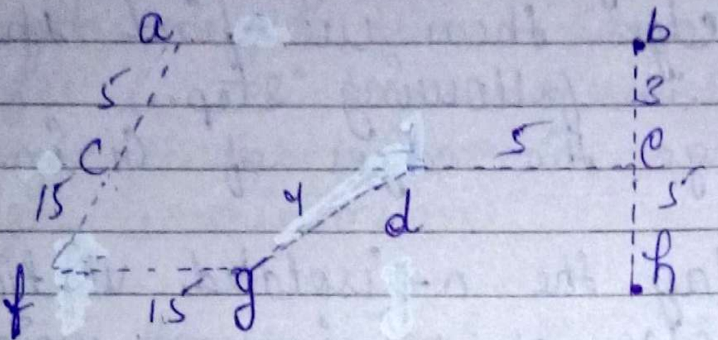
edges	(b,e)	(g,d)	(d,e)	(e,h)	(a,c)	(c,f) ~~(a,b)~~	(f,g)
	3	4	5	5	5	15	15

(a,d)	(g,h)	(a,b)	(c,g)
15	15	15	18

Teacher's Signature

Scanned by CamScanner

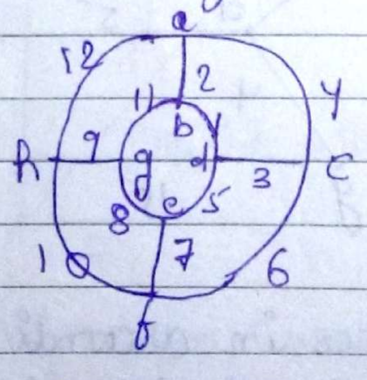
Step 2 → ~~Log~~ Locate the 8 vertices of graph.



Step 3 — Select smallest edge (b, c) first & make connection as shown.

Step 4 — Select second lowest edge (g, d) then (d, e) then (e, h) there (a, c) then (c, f) then (f, g) which cover all the vertices of G with out circuit. This gives a spanning tree with weight 52.

Ques → Construct a minimum spanning tree for the following weight graph.



Ans → In this case we will use Kruskal method.

Step 1: Arrange the edges in ascending order according to it weight.

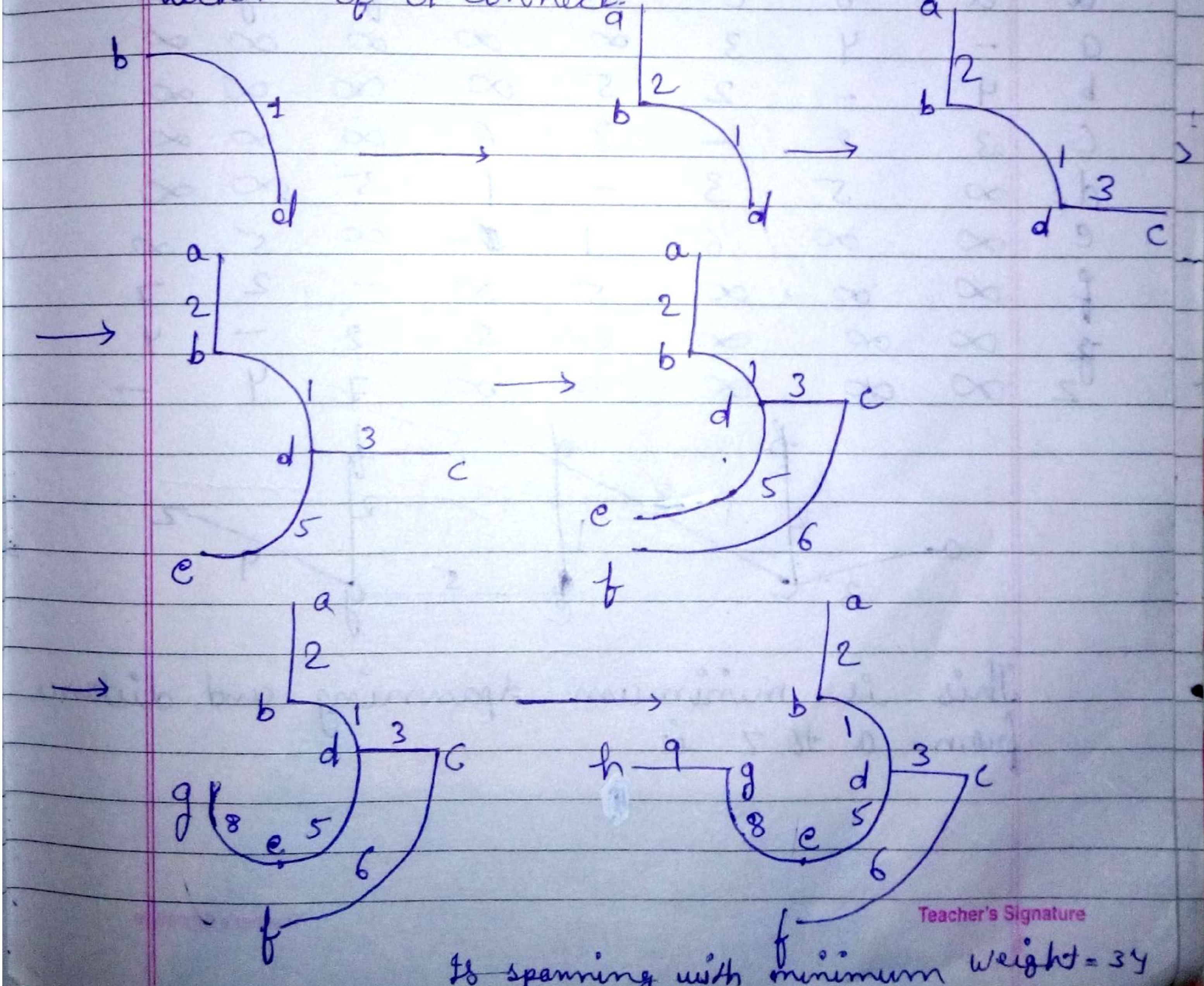
Scanned by CamScanner

DATE: / /
PAGE NO.

Edge	(b, d)	(a, b)	(c, d)	(a, c)	(d, e)	(f, c)	(e, f)
weight	1	2	3	4	5	6	7

edge	(g, e)	(g, h)	(h, f)	(g, b)	(b, a)
weight	8	9	10	11	12

Step 2. Display the isolated vertex a, b, c, d, e, f, g, h then start with minimum edge. Connect it between vertices then second minimum connect the vertex be such that no circuit will form continue the process till all the vertex of G connect.

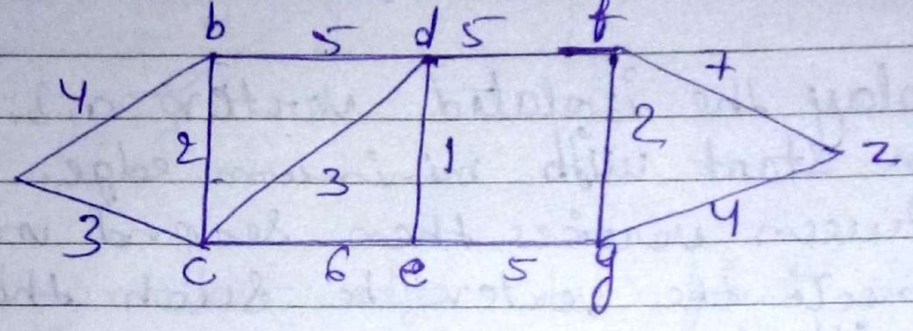


Is spanning with minimum weight = 34

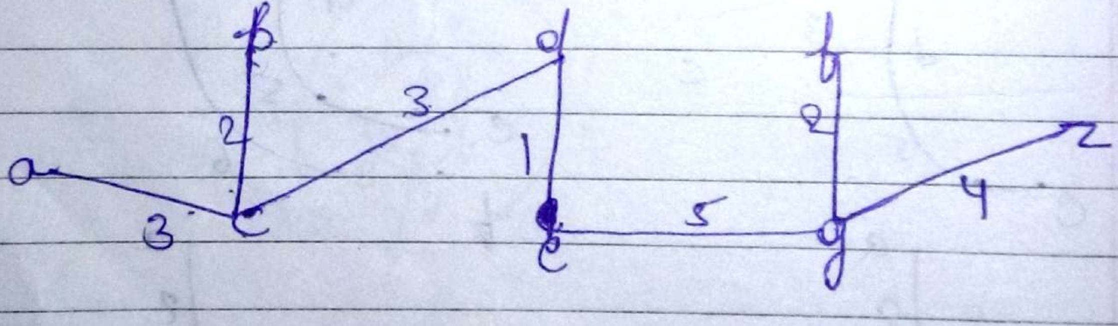
Teacher's Signature

(ii) Prim's Algorithm → Let G be a graph with n-vertices then we find spanning in following ~~series~~ Steps

Example



	a	b	c	d	e	f	g	z
a	–	4	3	∞	∞	∞	∞	∞
b	4	–	2	5	∞	∞	∞	∞
c	3	2	—	3	6	∞	∞	∞
d	∞	5	3	—	1	5	∞	∞
e	∞	∞	6	1	–	∞	5	∞
f	∞	∞	∞	5	∞	–	2	7
g	∞	∞	∞	∞	5	2	–	4
z	∞	∞	∞	∞	∞	7	4	—



This is minimum spanning and distance from a to z is

Scanned by CamScanner