Date

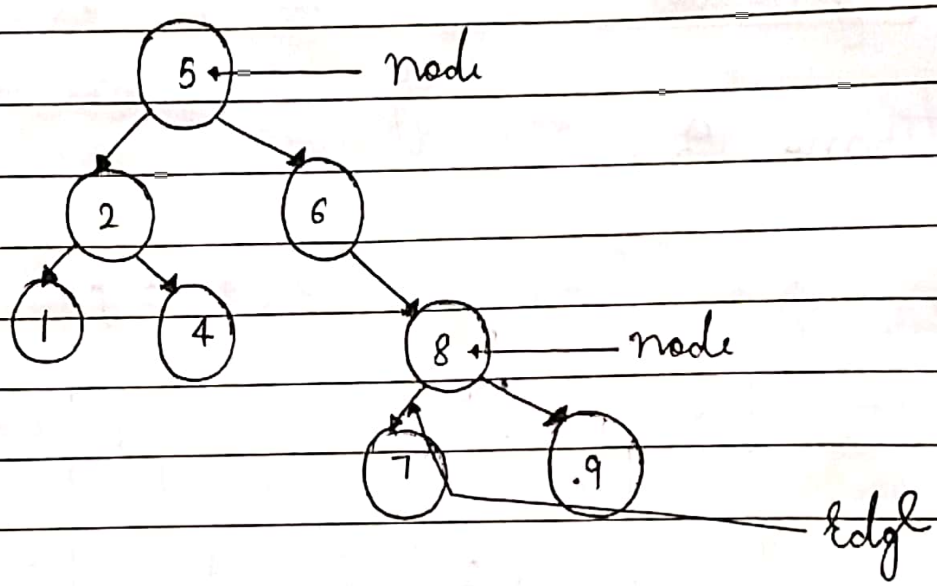
Page

UNIT-3

CHAPTER 3.1

TREE =

→ Tree Is A discrete structure that represents hierarchical relationships between Individual Elements Or nodes.



SIBLING =

→ A node with same parent.

ANCESTOR =

→ Parent node of a child node Is termed As Ancestor.

DESCENDENT =

→ All the hierarchical children of parent node Are called As the descendents of the parent node.

Date ________

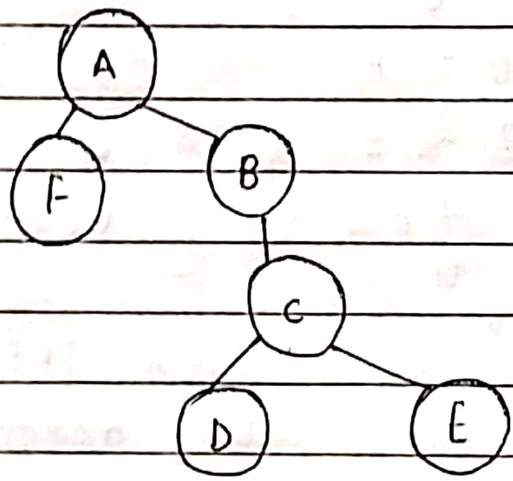
Page ________

LEAF =

→ A node with no children Is termed As leaf node.

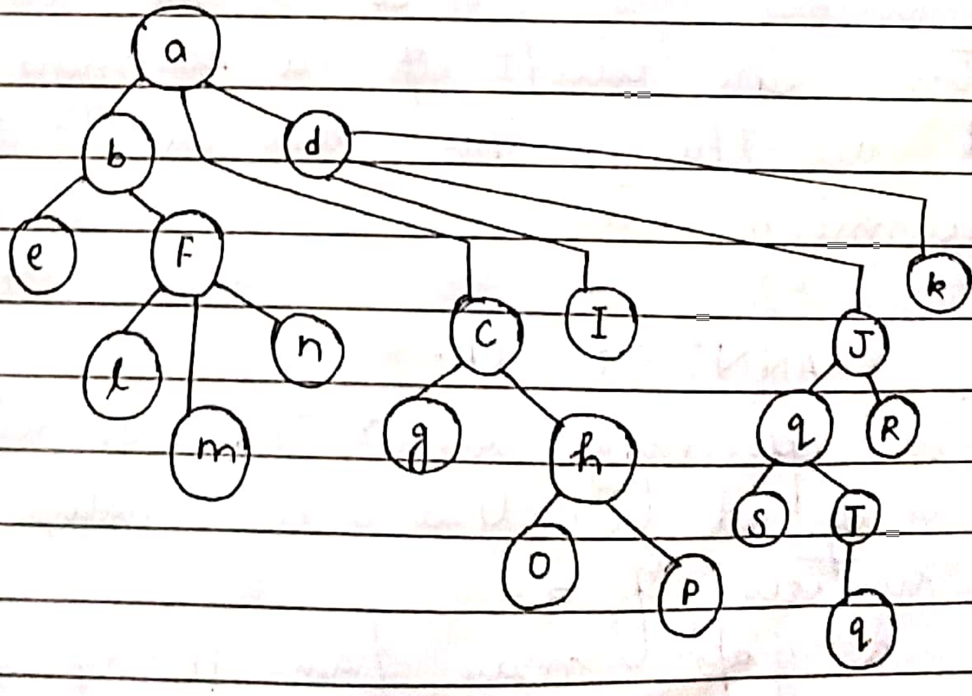
INTERNAL NODES =

→ Any node which Is not A leaf that Is termed As Internal nodes.



Here, Ae, F, B, C Are the Internal nodes.

Q.



(i) Which vertex Is root ?
→ a vertex Is the root.
(ii) Which vertices / vertex Are Internal nodes ?
→ b, c, d, F, g, h, J, q, S, t Are Internal nodes.
(iii) Leaf Nodes = e, l, m, n, g, o, P, S, q, R
(iv) Children of J = q, R
(v) Parent of H = C
(vi) Siblings of O = P
(vii) Ancestors of m = F, b, a
(viii) Descends of B = e, F, l, m, n

MINIMUM SPANNING TREE =

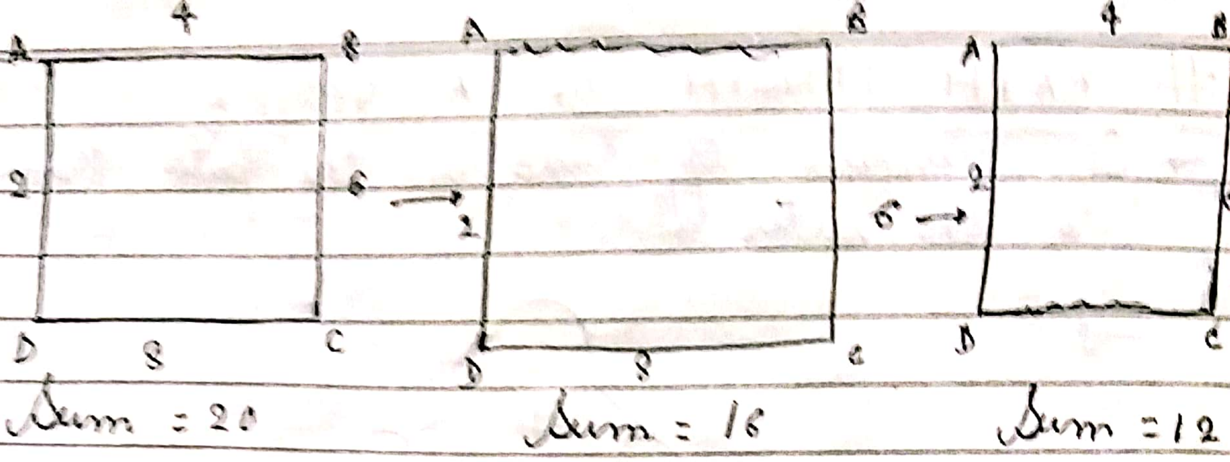
→ A spanning tree with Assigned weight less then Or Equal to the weight of Every possible spanning tree of a weighted, connected And Undirected graph G Is called minimum spanning tree. The weight of A spanning tree Is the sum of All the weights Assigned to each edge of spanning tree.

SPANNING TREE =

→ A spanning tree of the connected Undirected graph G Is A tree that minimally Includes All of the vertices of G.

→ A graph may have many spanning tree.

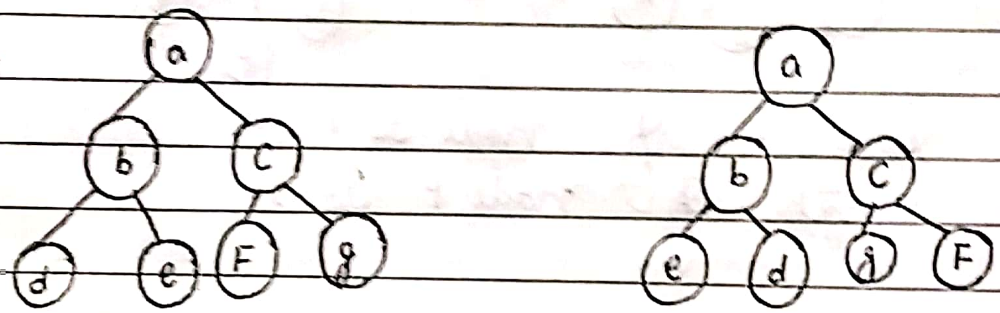
Eg :



ORDERED TREES :

→ If in a tree at each level an ordering is defined then such a tree is called an ordered tree.

Eg :



ROOTED TREES :

→ If a directed tree has exactly one node or vertex called root whose incoming degree is zero and all other vertices having incoming degree one then the tree is called rooted tree.

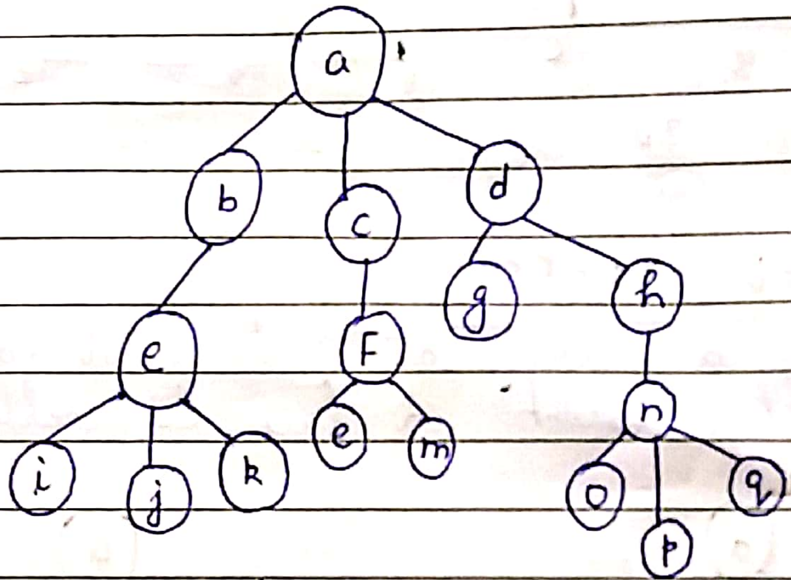
→ A tree with no node is also a rooted tree (Empty tree).

→ A single node with no children is called a rooted tree.

PATH LENGTH OF A VERTEX =

→ The number of edges in the path from the root to the vertex.

Eg =

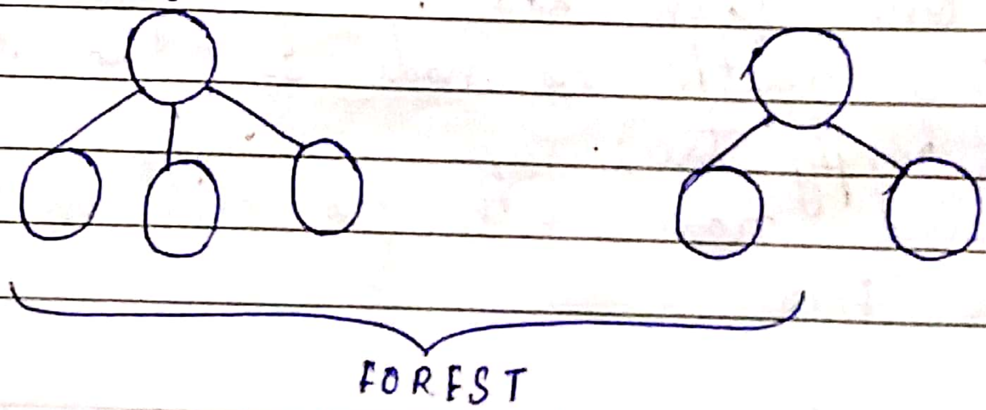


(i) Path length of node Is 1.
(ii) Path length of node F Is 2.

FOREST =

→ Let A be the root And the corresponding edges connecting the node Are deleted from A tree we Obtain A set of disjoint trees. This set of disjoint trees Is called A forest.

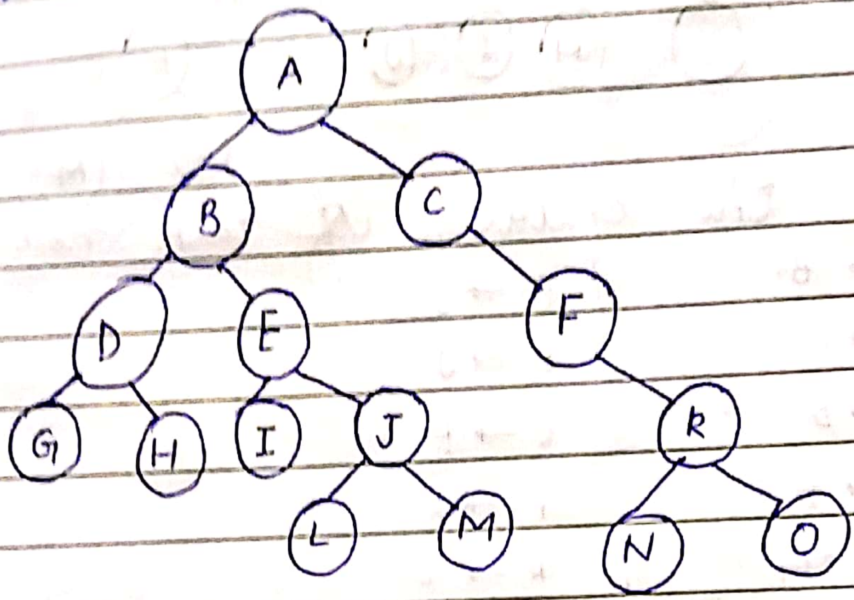
Eg =



Date
Page

BINARY TREE :

→ If the outdegree degree of every node is less than or equal to two then the tree is called binary tree.



Q (i) Which is the root = A

(ii) Which node are leaf = G, H, I, L, M, N, O

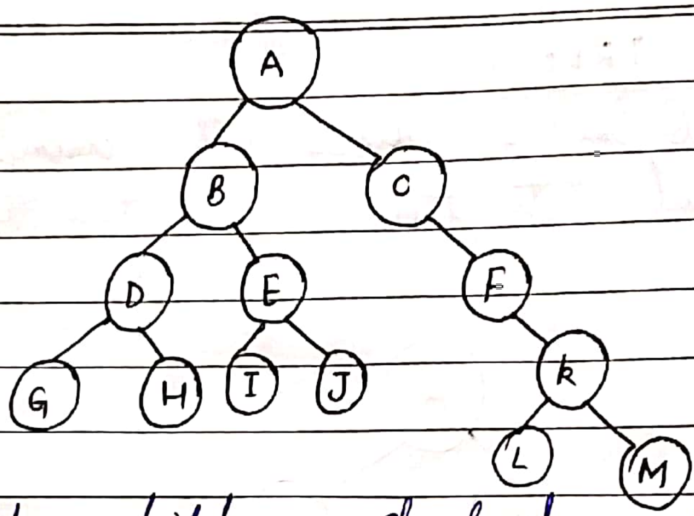
(iii) Name parent node of each node =

B → A	I → E
C → A	J → E
D → B	L → J
E → B	M → J
F → C	K → F
G → D	N → K
H → D	O → K

Scanned with CamScanner

Date
Page

Q1.



(i) List the children of each node

A → B
A → C
B → D
B → E
D → G
D → H
E → I
E → J
C → F
F → k
k → L
k → M

(ii) List the sibling

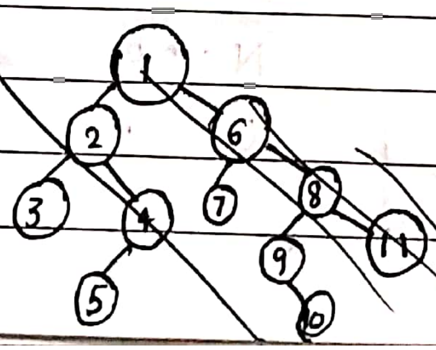
→ (B, C), (D, E), (G, H), (I, J), (L, M)

(iii) Find depth of each node

→ A → 0
B → 1
C → 1
D → 2
E → 2
F → 2
G → 3
H → 3
I → 3
J → 3
k → 3
L → 4
M → 4

Max. depth = 4
Max level = 4

Q.

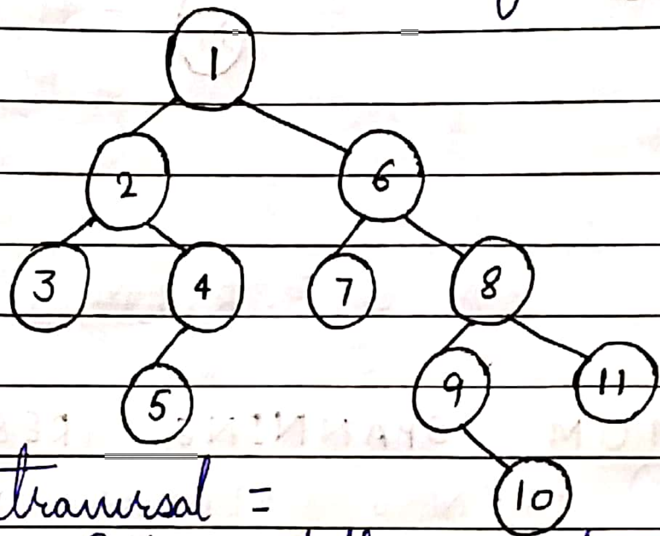


TRAVERSING IN TREE =

→ Traversing means to visit All the nodes of the tree

(i) Pre - Order traversal = Root → Left → Right
(ii) Post - Order traversal = Left → Right → Root
(iii) In - Order traversal = Left → Root → Right

Q.



Pre - Order traversal =
Root → Left → Right

⇒ 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11

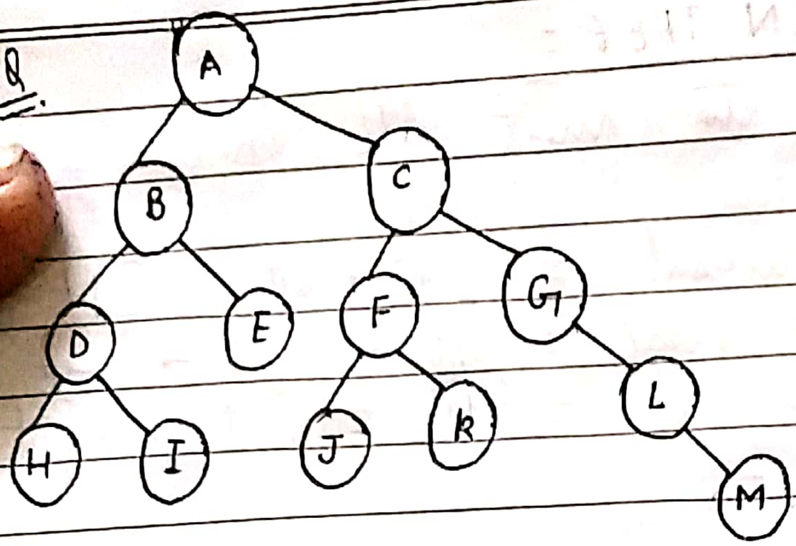
Post - Order traversal = ~~R~~ Left → Right → Root

⇒ 3, 5, 4, 2, 7, 10, 9, 11, 8, 6, 1.

In - Order traversal = Left → Root → Right

⇒ 3, 2, 5, 4, 1, 7, 6, 9, 10, 8, 11

Date
Page

Q. 

Pre - Order =
R – L – R
A, B, D, H, I, E, C, F, J, K, G, L, M

Post Order =
L – R – R
H, I, D, E, B, J, K, F, M, L, G, C, A

In - Order =
L – R – R
H, D, I, B, E, A, J, F, K, ~~L, M~~ C, G, L, M.

MINIMUM SPANNING TRESS ALGORITHM =

(i) PRIM'S ALGORITHM
(ii) KRUSKAL'S ALGORITHM

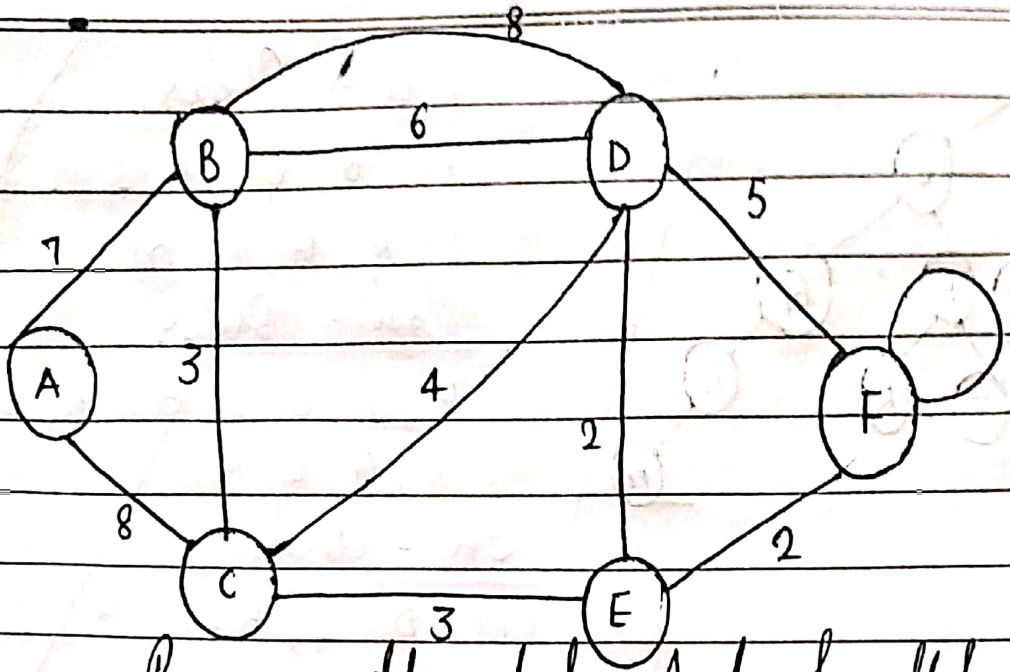
Note =
- In MST $V = V'$ and $E' < E$
- Normal $G(V, E)$, MST $G'(V', E')$
- $\boxed{E' = |V| - 1}$

* PRIM'S ALGORITHM =

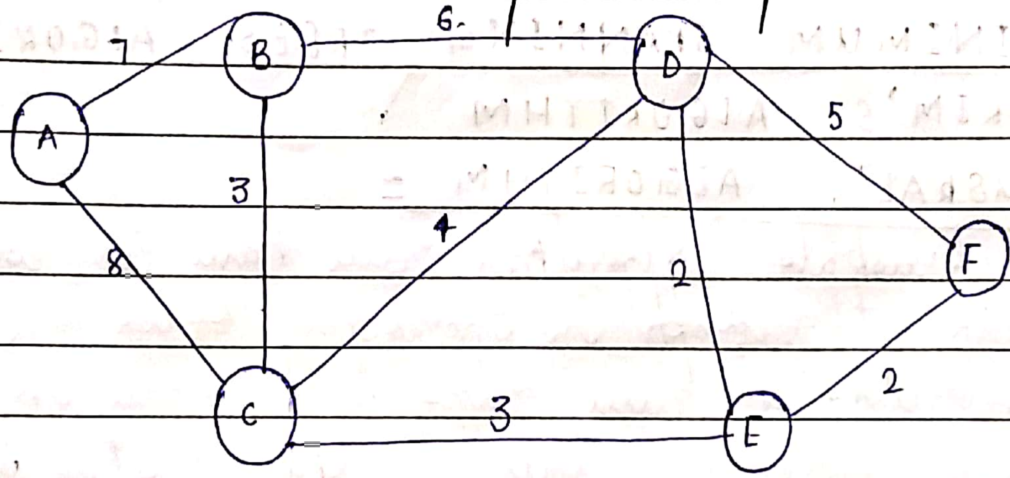
→ Prim's Algorithm Is A greedy Algorithm that Is Used to find the MST from A graph.
→ It starts with A single node And Explore All the Adjacent node with All the connected Edges At Every step.

Scanned with CamScanner

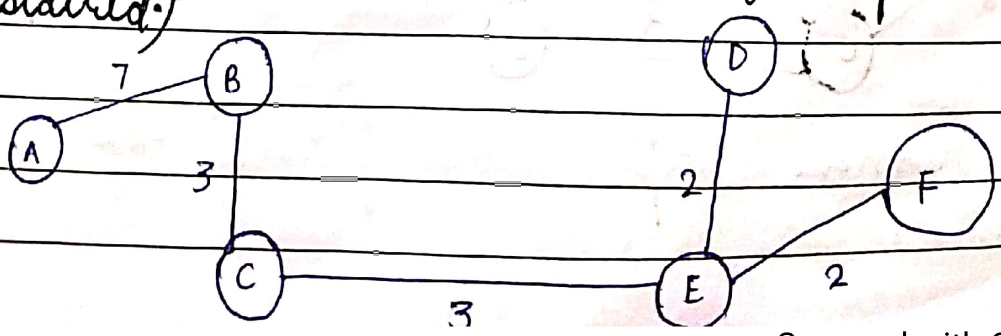
Q =



Step 1 = Remove the loop and parallel edges.



Step 2 = Choose any vertex as the root node and check all the outgoing edges from this node. (By comparing all the previous weighted edges, the edge which has minimum weight without making loop will be considered.)



$G'(V') = 6$

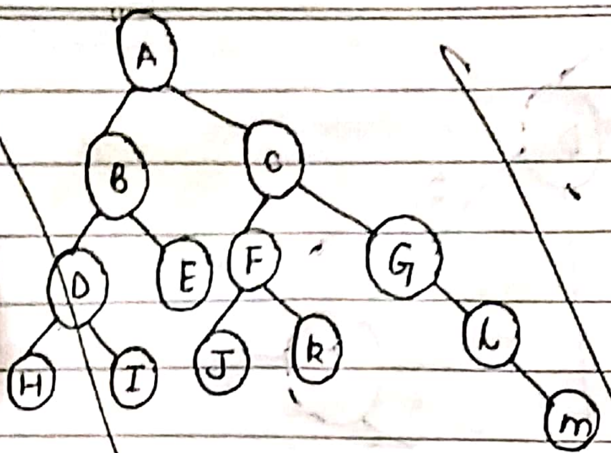
$G'(E') = 5$

$\therefore \boxed{G'(V') - 1 = G'(E')}$

Date ______

Page ______

Q



Pre - Order

A, B, D, H, I, E, C, F, J, k, G, l, m

Post - Order

H, I, D, E, B, J, k, F, m, l, G, C, A

In - Order

H, D, I, B, E, A, J, F, k, C, G, l, m

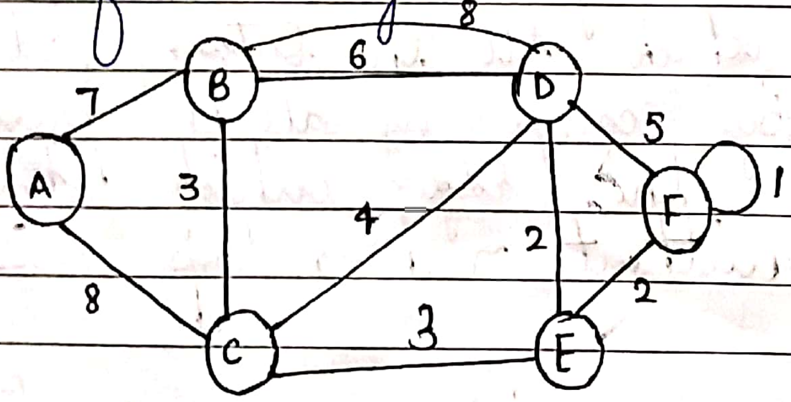
* MINIMUM SPANNING TREES ALGORITHM =

(i) PRIM'S ALGORITHM

* KRUSKAL'S ALGORITHM =

→ In kruskal's Algorithm we have to sort all the Edges of the graph In Increasing Order. Then It keeps On Adding new Edges And nodes In the minimum spanning tree If the newly Added Edge does not form a cycle.

Q



Step 1 = Remove loops And parallel Edges

12 → IOT → 306 — 4:30 — 11

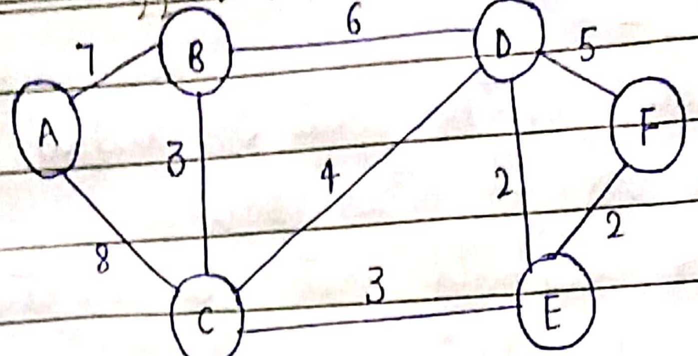
18 → Soft Skills → 110 → 3 Date 4:30

18 → CC → 303 → 11:15 — 12:45

19 → CC → 303 → 11:15 — 12:45 Page 4:30

20 → MAD → 309A → 3:00 — 4:30

21 → DM → ~~303 → 9:30 — 11:00~~



EF = 2

DE = 2

CE = 3

BC = 3

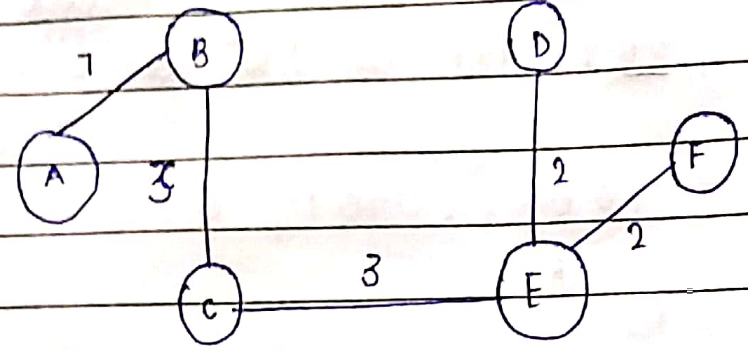
CD = 4

DF = 5

BD = 6

AB = 7

AC = 8



$E' = 5$

$V' = 6$

$\therefore \boxed{G(E') = G(V') - 1}$

PROPERTIES OF MST (Minimum Spanning Tree)

(i) The disconnected graph does not have spanning tree.

(ii) A complete Undirected graph can have n^{n-2} number of spanning trees where n is the number of vertices.

(iii) Adding One Edge to spanning tree will create a loop.

(iv) The spanning tree must not form a cycle i.e., no Edge Is traversed twice.

(v) If Each Edge has a distinct weight then there will be Only a Unique minimum spanning tree.

*. PREFIX CODE =

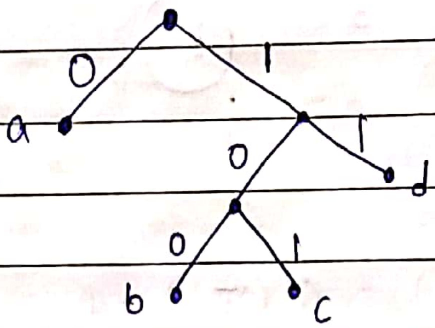
→ It Is A variable length code In which no code word Is A prefix of Another One.

Should not be repeated

Q. A = 00, 11, 110, 111 ✓ Prefix code

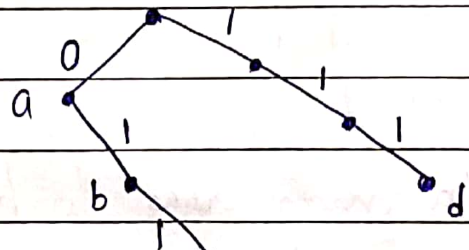
B = 00, 001, 111 × Not A prefix code

Q. (i) a = 0, b = 100, c = 101, d = 11



✓ Prefix code

(ii) a = 0, b = 01, c = 011, d = 111



× Not a prefix code (because a, b and c lies on same trajectory.)

HUFFMANN CODING =

→ It Is A technique of compressing the data to reduce Its size without loosing Any of the detail.

→ It was first developed by David Huffman.

→ It Is Useful to compress the data In which there

Date

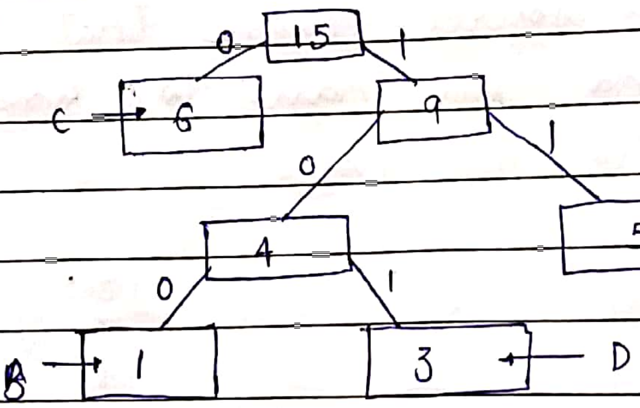
Page

Are frequently Occurring characters.

Note: It Is basically Used to compress long bits to short bits.

Q. B C A A D D D C C A C A C A C

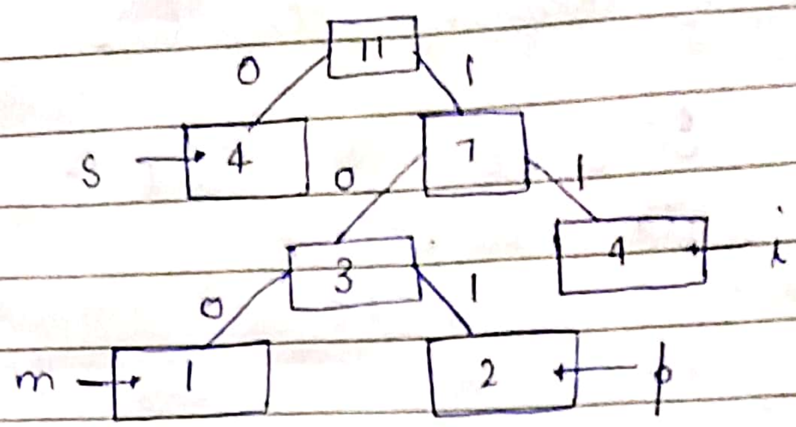
Char	freq	code	size = frequency count * code length
A	5	11	5 X 2 = 10
B	1	100	1 X 3 = 3
C	6	0	6 X 1 = 6
D	3	101	3 X 3 = 9
4 X 8 = 32 bits	15 bits		28 bits



Q. How many bits are required for Encoding the message 'mississippi'? Below Is the frequency count.

Char	freq	Code	Size
m	1	100	1 X 3 = 3
i	4	11	4 X 2 = 8
s	4	0	4 X 1 = 4
p	2	101	2 X 3 = 6
4 X 4 = 16 bits	11 bits X 8 = 88 bits		~~32 + 11~~ bits

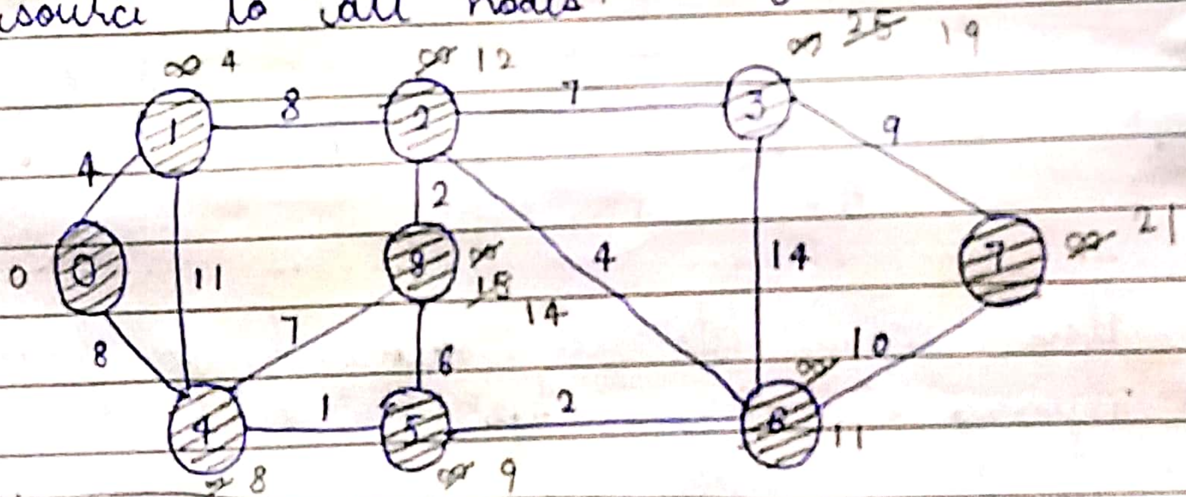
Date
Page

→ 

SHORTEST DISTANCE IN WEIGHTED GRAPHS:

(i) DIJKSTRA'S ALGORITHM (Single source shortest path)

→ It Is A single source shortest path Algorithm.
& Here, single source means that Only One source Is given And we have to find the shortest path from source to All nodes.

Q 

$$\boxed{d[u] + c[u, v] < d[v]}$$

Let, u = 0, v = 1

∴ d[u] = 0, c[u, v] = 4, d[v] = ∞

∴ A/g,

0 1 4 ∞ ∞

1 ∞ ∞

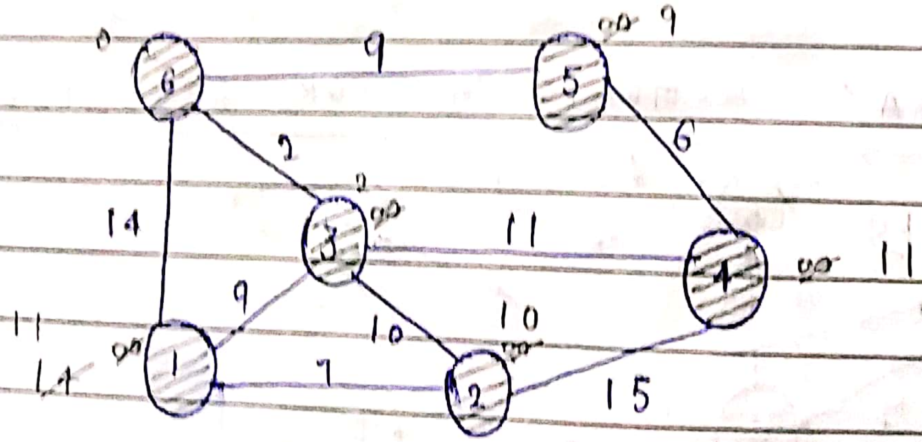
∴ | d[v] = 4 |

| d[1] = 4 |

III) for every vertex and make them visited

	0	1	2	3	4	5	6	7	8
0	0	4	12	19	8	9	11	21	14
1	∞	0	8	15	∞	∞	∞	∞	∞
2	∞	∞	0	7	∞	∞	∞	∞	2
3	∞	∞	∞	0	∞	∞	∞	∞	∞
4	∞	∞	∞	∞	0	1	3	13	7
5	∞	∞	∞	∞	∞	0	2	12	6
6	∞	∞	∞	∞	∞	∞	0	10	∞
7	∞	∞	∞	∞	∞	∞	∞	0	∞
8	∞	∞	∞	∞	∞	∞	∞	∞	0

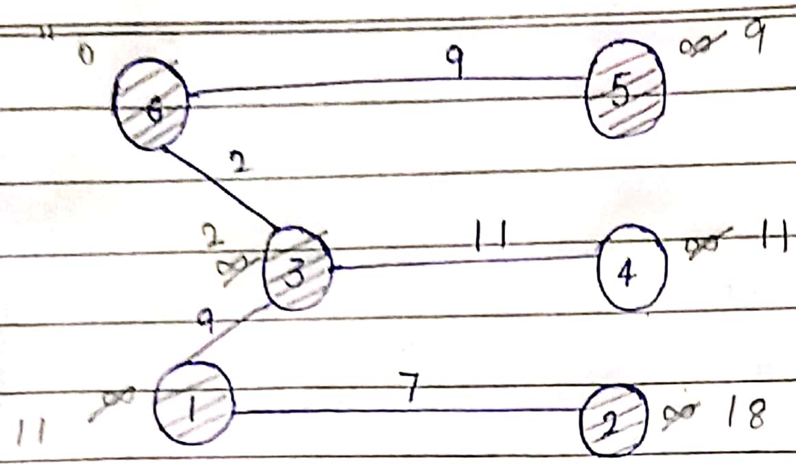
Q.



Date ________

Page ________

Q.

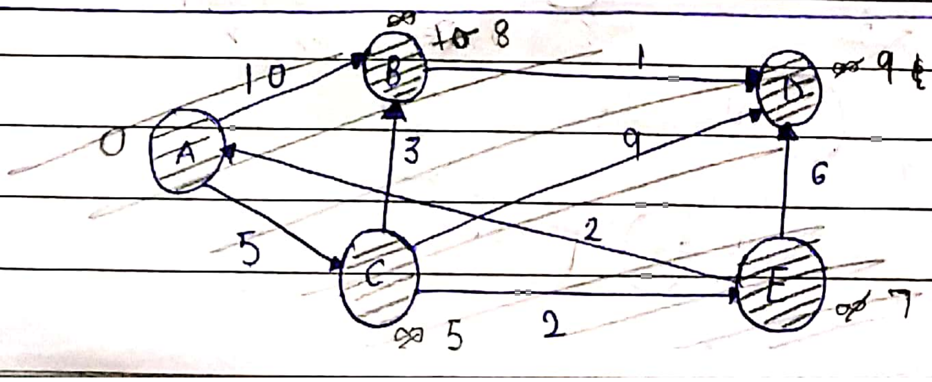


Source	Destination				
6	1	2	3	4	5
0	∞	∞	∞	∞	∞
0	∞	∞	2	∞	9
0	11	∞	2	11	9
0	11	∞	2	11	9
0	11	18	2	11	9
0	11	18	2	11	9

Note = A node which Is Already Used As source for some Other node cannot be Updated.

DIJKSTRA ALGORITHM FOR DIRECTED GRAPH

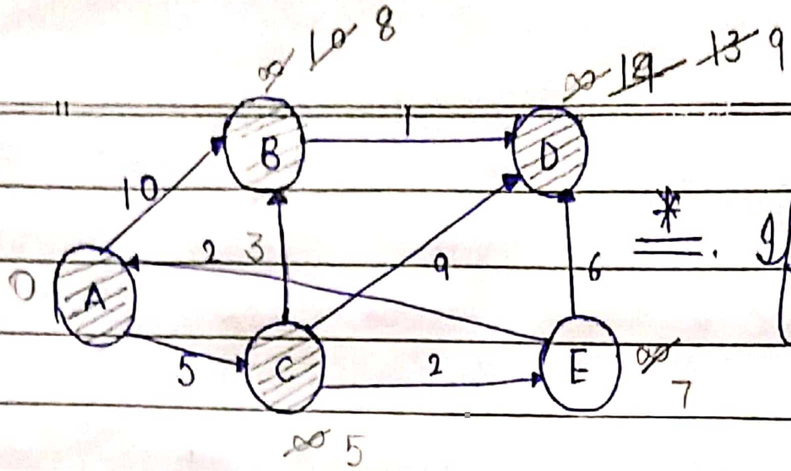
Q.



FREEMIND

Date

Page



Source vertex = A

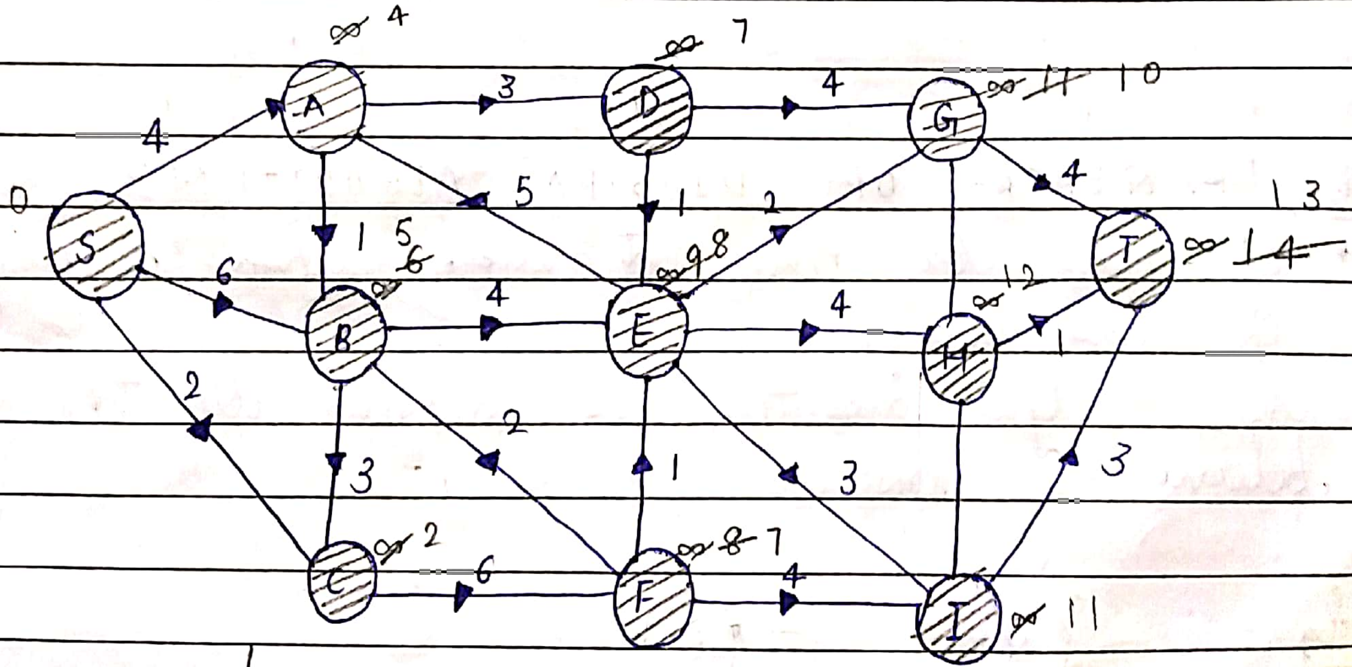
*. If $d[u] + c[u,v] < d[v]$

then

$d[v] = c[u,v] + d[u]$

	A	B	C	D	E
A	$\boxed{0}$	∞	∞	∞	∞
C		10	$\boxed{5}$	∞	∞
E		8		14	$\boxed{7}$
B		$\boxed{8}$		13	
D				$\boxed{9}$	

Q



	S	A	B	C	D	E	F	G	H	I	T
S	$\boxed{0}$	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
C		4	6	$\boxed{2}$	∞	∞	∞	∞	∞	∞	∞
A		$\boxed{4}$	6		∞	∞	8	∞	∞	∞	∞
B			$\boxed{5}$		7	9	8	∞	∞	∞	∞

	S	A	B	C	D	E	F	G	H	I	T
D					$\boxed{7}$	9	7	∞	∞	∞	∞
F						8	$\boxed{7}$	11	∞	∞	∞
E						$\boxed{8}$		11	∞	11	∞
G								$\boxed{10}$	12	11	∞
I									12	$\boxed{11}$	14
H									$\boxed{12}$		14
T											$\boxed{13}$

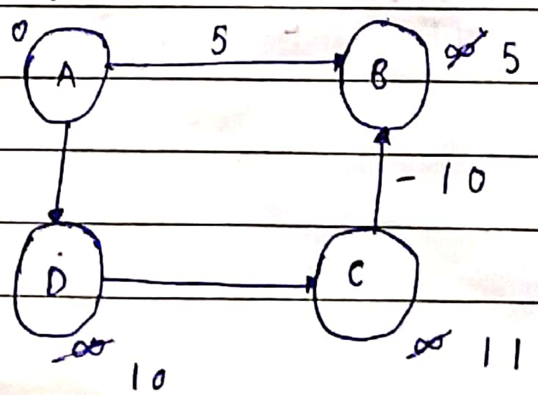
S → C → A → B → D → F → E → G → I → H → ...

* DRAWBACK OF DIJSTRA'S ALGORITHM :

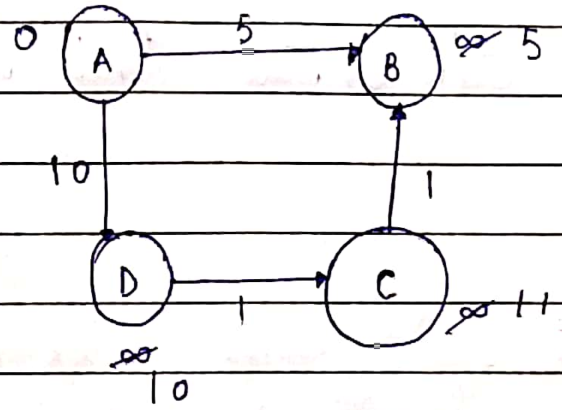
→ It may or may not work when the weight of Edges are negative.

→ As, this Algorithm Is designed On the basis of positive (+ve) weight.

Eg =



→ Suppose we want to Update the distance from vertex C(U) to vertex B(V) = 11-10 = 1
→ But we cannot Update It because this node Is Already visited.
So, In this case dijkstra Algorithm has given wrong distance.
This Is not Always true.
→ Suppose If we change the value As -1 from vertex C(U) to vertex B(V) = 11 - 1 = 10

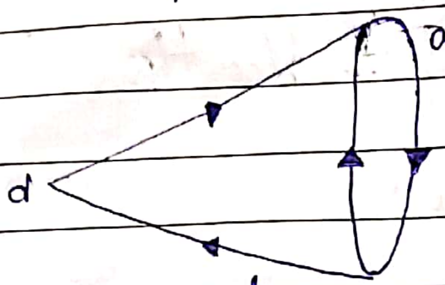


GRAPH

→ The graph consists Of points Or nodes called vertices which Are connected to Each Other by way Of lines called Edges.
→ These lines may be directed Or Undirected.

(i) DIRECTED GRAPH =
→ A directed graph Is defined As An Ordered pair (V, E) where V Is A set And E Is A binary

relation On V :
→ The Elements In V Are called vertices.
The Ordered pair In E Are called Edges.



(directed graph)

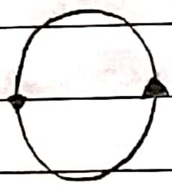
→ Here the vertices Are (a, b), (b, a), (b, d), (d, a).

→ The vertex a Is called the Initial And vertex D Is called the terminal

② LOOP IN GRAPH :

→ An Edge that Is Incident from same vertex And has reflection On same vertex Is called loop a self-loop.

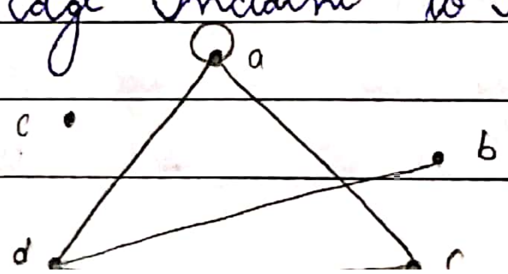
→ Eg -



(loop)

③ ISOLATED VERTEX :

→ A vertex Is said to be Isolated vertex If there Is no Edge Incident to It.



Date
Page

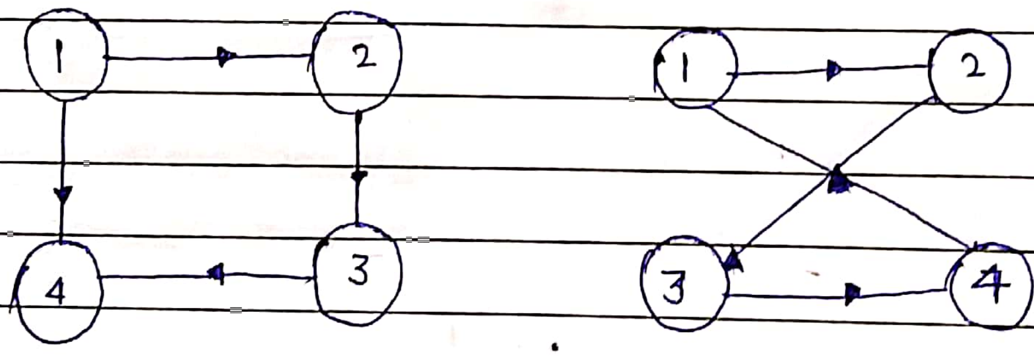
→ Here vertex 'e' Is An Isolated vertex.
→ Here, a Is having self loop.
→ Here, b Is a Pendent vertex.
that has Only One Edge Incident to It.

④ UNDIRECTED GRAPH =

→ It consists of A set of vertices V And A set of Edges E. The Edge set contains the Unordered pairs of vertices.

Eg = Let V = {1, 2, 3, 4} And E = {(1, 2), (1, 4), (3, 4), (2, 3)}

The graph can be drawn In various ways One of which Is



•) Mixed Graph = It Is the mixture of both directed And Undirected graph.

•) Odd vertex = degree Odd

•) Even vertex = degree Even

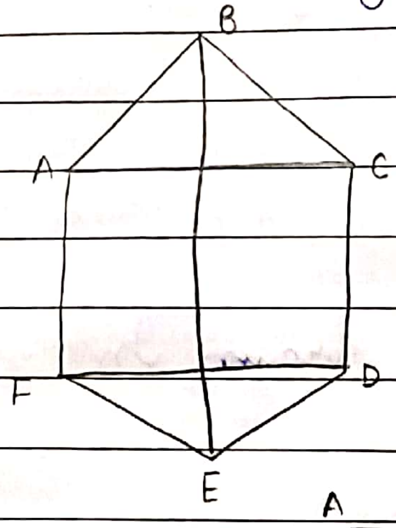
SUBGRAPH =

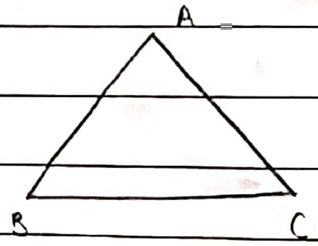
→ A subgraph of A graph G = (V, E) Is A graph

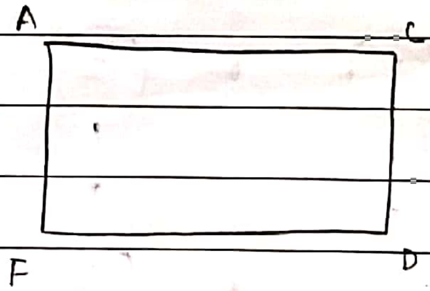
Scanned with CamScanner

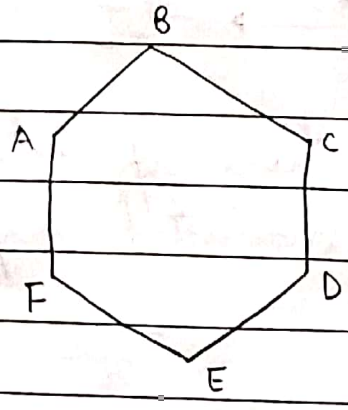
$G' = (V', E')$ In which $V' \subseteq V$ And $E' \subseteq E$ And each edge of G' And G has same end vertices In G' As In graph G.

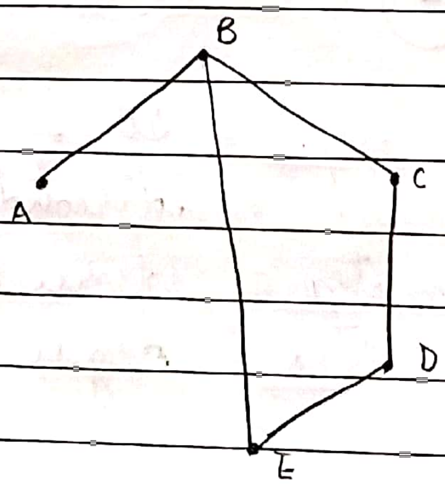
Q. Consider the graph G As shown below. Show the subgraphs In the given graph.









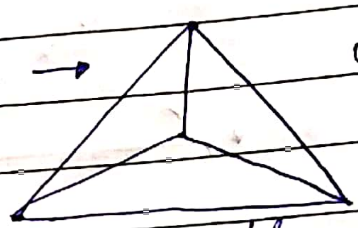


COMPLEMENT OF A SUBGRAPH :-

→ Let $G = (V, E)$ be a graph and S be a subgraph of edges of S be deleted from graph G, the graph so obtained is the complement of subgraph S. It is denoted by $\bar{S}$.

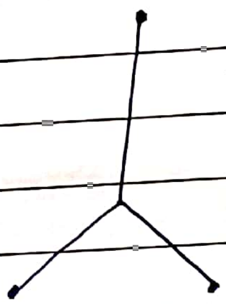
$\therefore \boxed{\bar{S} = G - S}$

Q. Consider the graph $G \rightarrow$



and its sub-graph

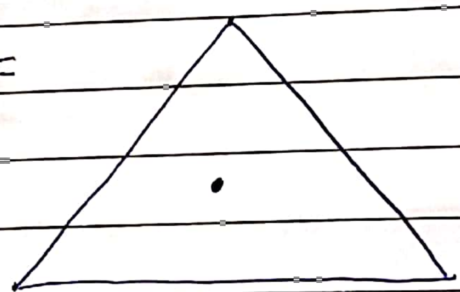
$S \rightarrow$



then create the compliment of subgraph S.

→ $\bar{S} = G - S$

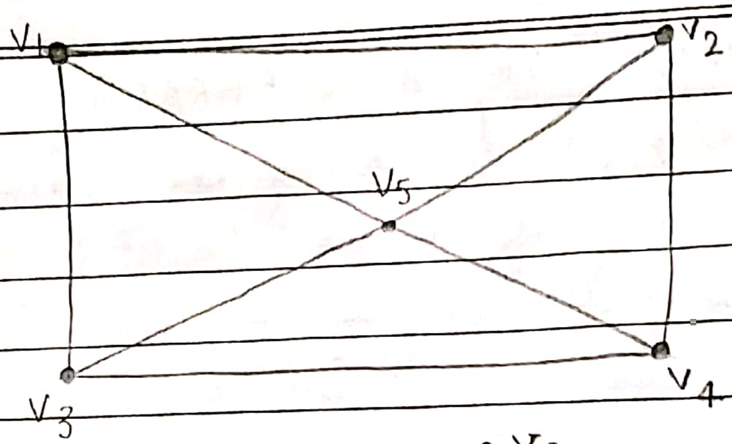
So, $\bar{S} =$



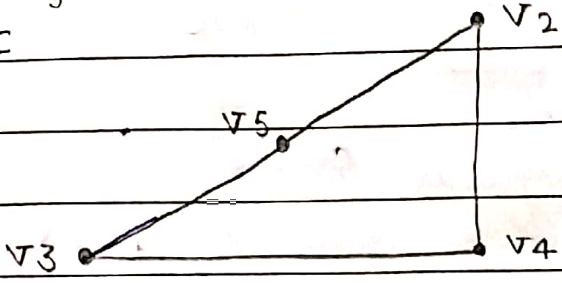
Note :- The number of vertices do not change in the compliment of a subgraph.

Q. Consider the graphs shown below. Determine the ~~compliment of a~~ subgraph.

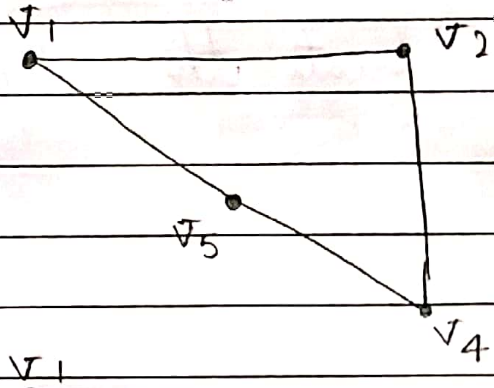
(i) G =



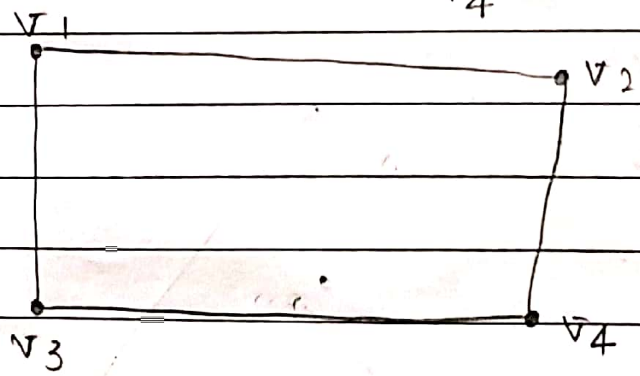
(i) G - V1 =



(ii) G - V3 =



(iii) G - V5 =



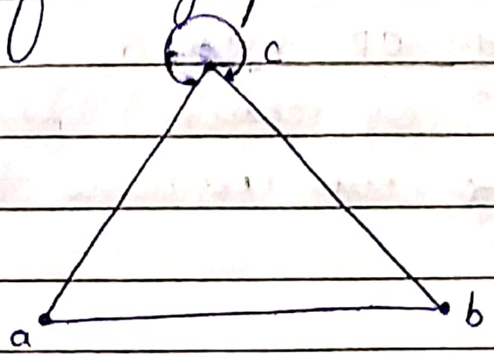
ORDER OF A GRAPH =

→ Let G be a graph then the number of the vertic In the graph G Is called Order of a graph.

SIZE OF A GRAPH =

→ Let G be a graph then the number of Edges In A group Is called the size of A graph.

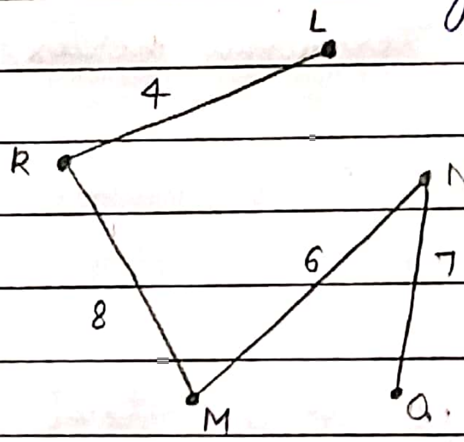
Q. Consider the graph As shown below. Determine the =
(i) Order of A graph = 3 (ii) Size of a graph = 4



WEIGHTED GRAPH =

→ A graph G = (V, E) Is called A weighted graph If each Edge of graph G Is Assigned A positive number W called the weight of the Edge.

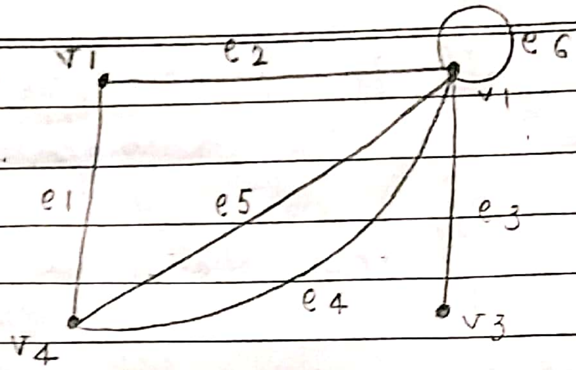
Eg =



MULTIGRAPH =

→ A multigraph G = (V, E) consists of A set of vertices V And A set of Edges E such that Edge set E may contain multiple Edges And

Eg :-



EULER'S PATH OR CHAIN

→ An Euler path (Or chain) through A graph is A path whose Edge list contains Each Edge of the graph Exactly Once.

Note :- No vertex Is visited twice.

* EULER'S CIRCUIT OR CYCLE

→ An Euler circuit (Or cycle) Is A path through A graph In which Initial vertex Appears second time As the terminal vertex.

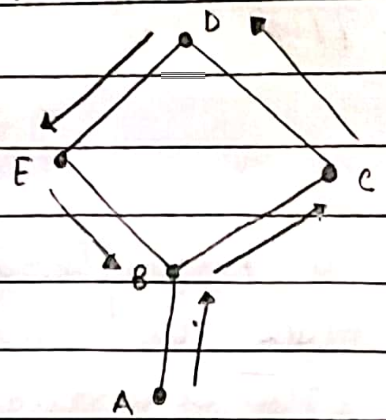
Note :- Start And End vertex Are same.

EULER GRAPH :-

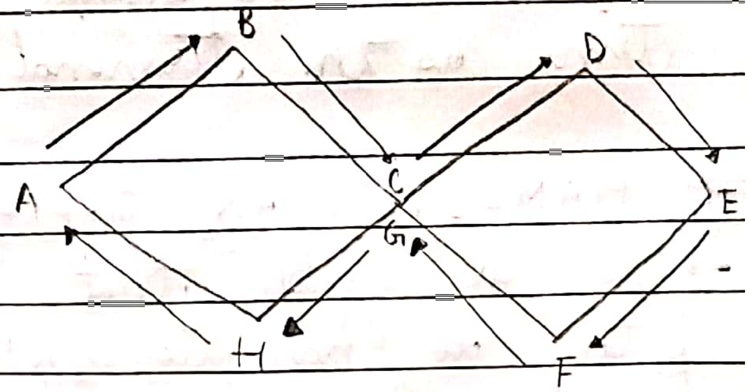
→ Euler's graph Is A graph that passes An Euler's circuit. Euler's circuit Uses Every Edge Exactly Once but vertex may be repeated.

Q. Consider the graph as shown below. Determine Euler's path, Euler's circuit and Euler's graph of them.

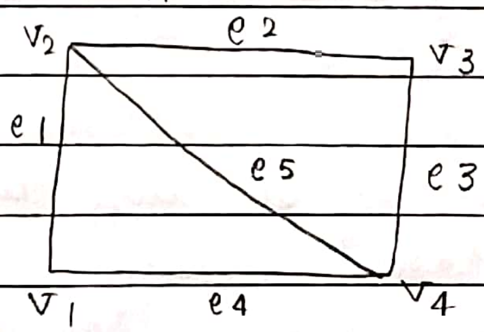
(a)



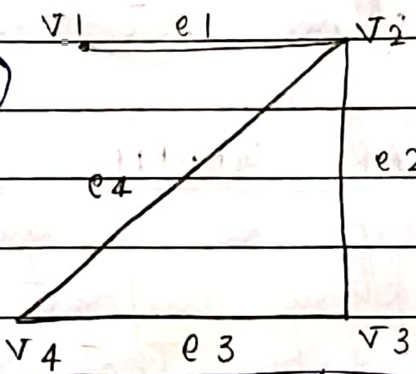
(b)



(c)



(d)



	(a)	(b)	(c)	(d)
Euler's path	Yes	Yes		
Euler's Circuit	No	Yes	Yes	No
Euler's graph	No	Yes	Yes	No

HAMILTONIAN PATH (OR CHAIN) :-

⇒ A hamiltonian path (or chain) through a graph is a path vertex list contain each vertex of the graph exactly once, ~~ex~~ except path is a circuit.

Date
Page

HAMILTONIAN CIRCUIT (OR CYCLE) :

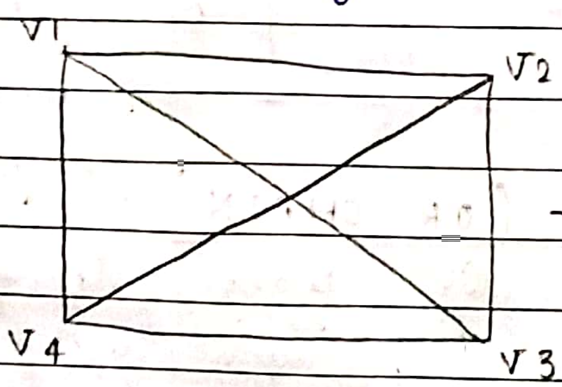
→ A hamiltonian circuit (Or cycle) Is A path In which the Initial vertex Appears A second time as the terminal vertex.

HAMILTONIAN GRAPH =

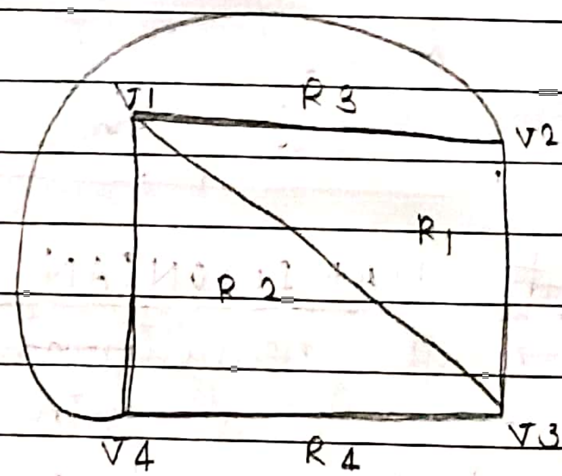
→ Is A graph that possesses A hamiltonian path. A hamiltonian path Uses each vertex Exactly Once but Edges may not be Included.

PLANAR GRAPH =

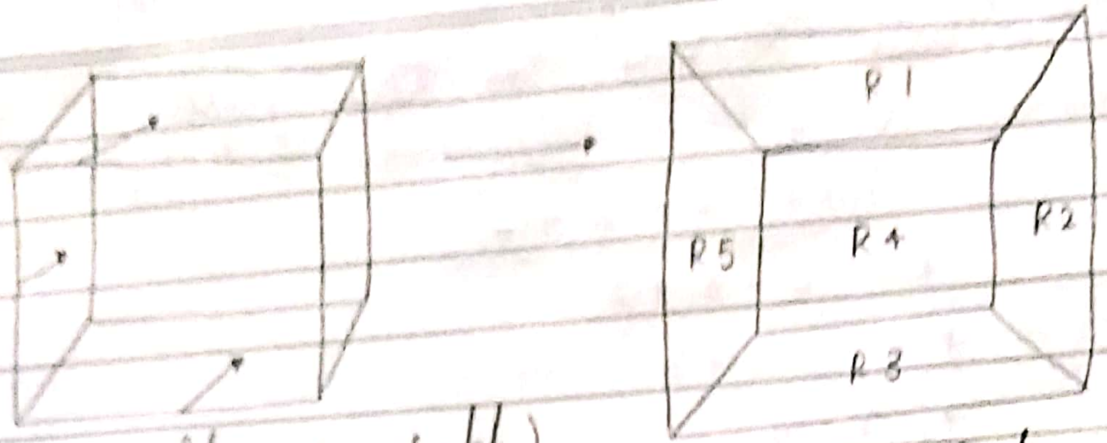
→ A graph Is called planar If It can be drawn In the plane without Any Edges crossing. And A graph that cannot be drawn On A plane without A crossover between Its Edges Is called non-planar graph.



(Non-planar graph)



R4 → Outer region
(Planar Graph)



(Non-planar graph) (Planar graph)

R6 → Outer regions

Theorem = A complete graph of 5 vertices Is non-planar.

COMPLETE GRAPH =

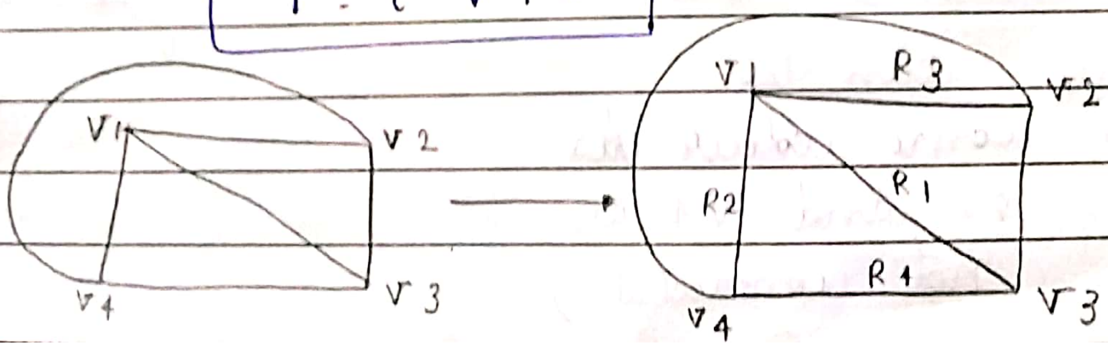
→ Edge between every vertex.
→ All Edges should be connected.
→ We will connect v_1 to v_4 And v_1 to v_3 So that Intersection should not be there.

* EULER'S FORMULA =

→ Let G be a connected planar graph with 'e' edges And vertices 'v'. Let 'r' be the number of regions In planar representation of G. Then,

number of regions

$$\boxed{r = e - v + 2}$$

→ 

(i) First draw regions for this graph

(ii) $e = 6, \; v = 4$

$$\therefore \boxed{r = e - v + 2}$$

$$r = 6 - 4 + 2$$

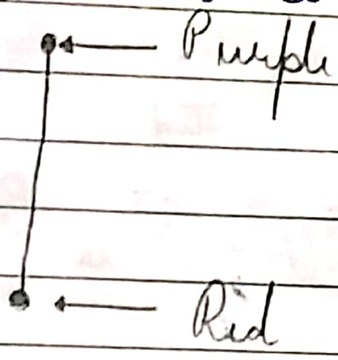
$$\Rightarrow r = 2 + 2$$

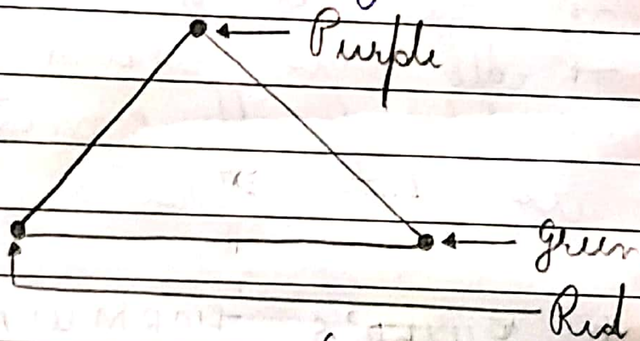
$$\Rightarrow \boxed{r = 4}$$

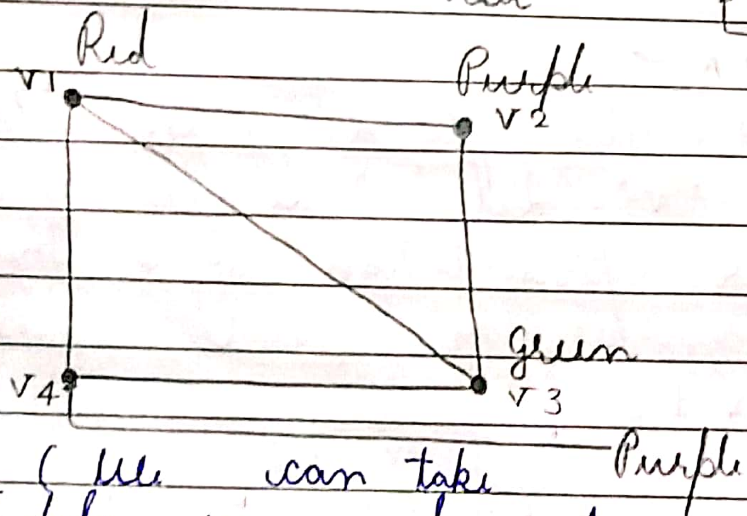
GRAPH COLORING AND CHROMATIC NUMBER:

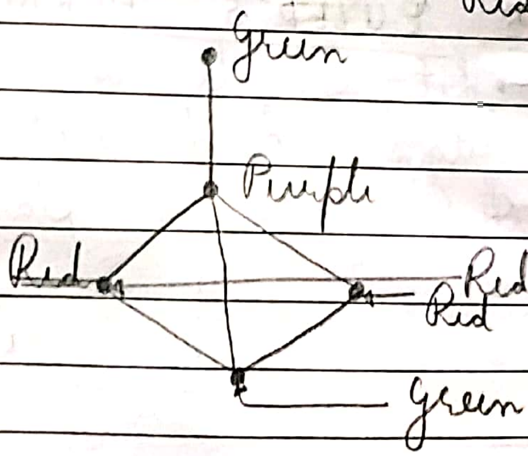
→ Painting all the vertices of a graph with colours such that no two adjacent vertices have the same color is called coloring of graph.

Eg =









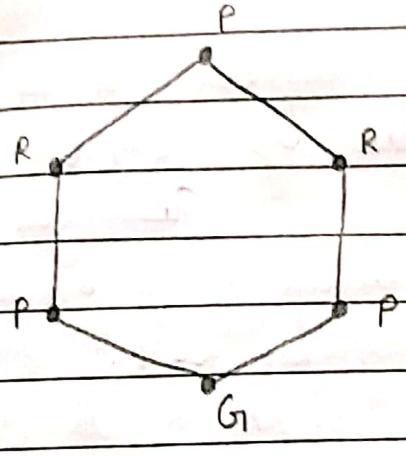
(We can take ~~(the~~ same colour as V2 and V4 are not connected.)

Note =

If a graph is circuit with n vertices then
(i) It is 2-chromatic if n is Even.
(ii) It is 3-chromatic if n is Odd.

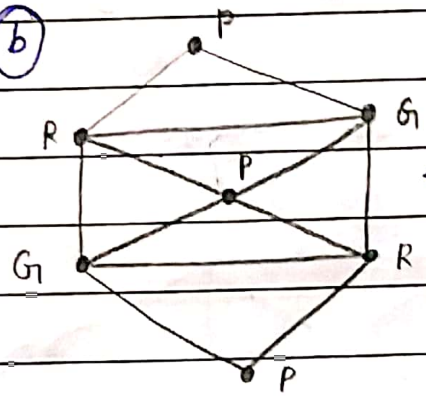
Q Determine the chromatic number of graph =

(a)



There are Even number of vertices.
Hence, chromatic number is 2.

(b)

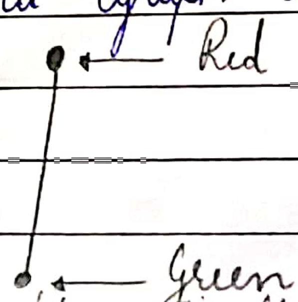


The number of vertices is Odd.
Hence, chromatic number is 3.

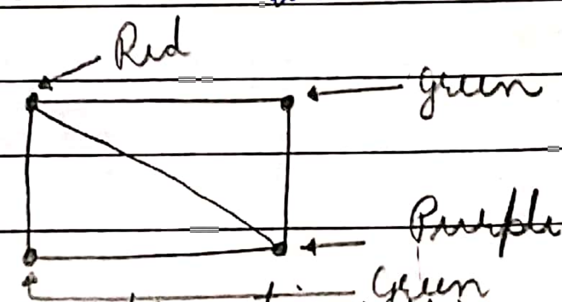
CHROMATIC NUMBER =

→ The least number of colors required for coloring of a graph G is called is chromatic number.

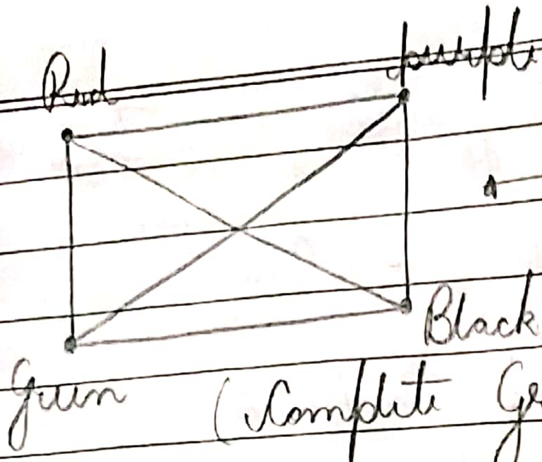
Eg =



(2 - chromatic graph)



(3 - chromatic graph)



→ 4 chromatic Graph

(Complete Graph)

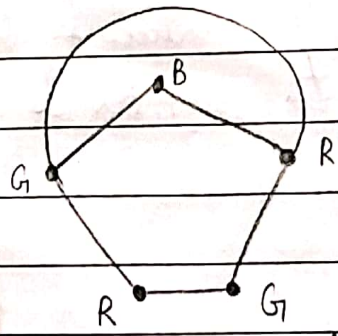
Note : • The chromatic number of a graph G is denoted by $\chi(G)$ ← (zai).

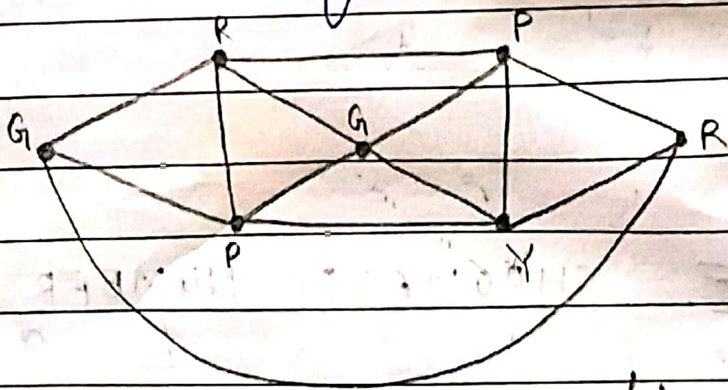
• If $\chi(G) = k$, the graph is called k-chromatic.

• Chromatic number of null graph (having only 1 vertex) is 1.

• Chromatic number of complete graph k of n vertices is n.

Q. Find the chromatic number of graph.



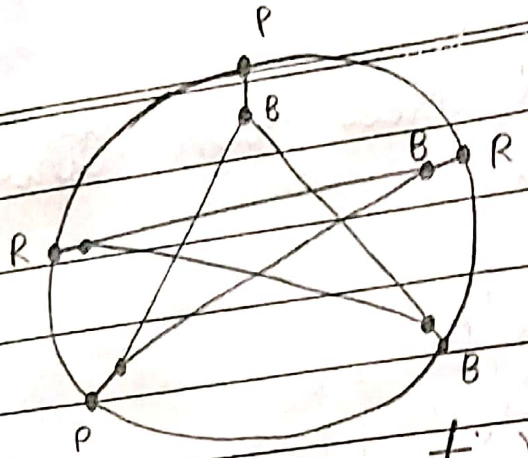


(3 - Chromatic)

→ Colour (G, B, R)

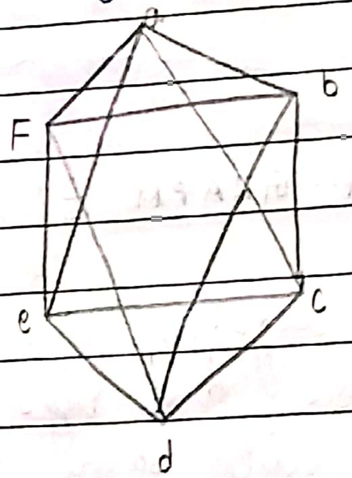
→ Odd number of vertices

4 - Chromatic



(3 - Chromatic) — Number of Unique colours Used
(R, B , P)

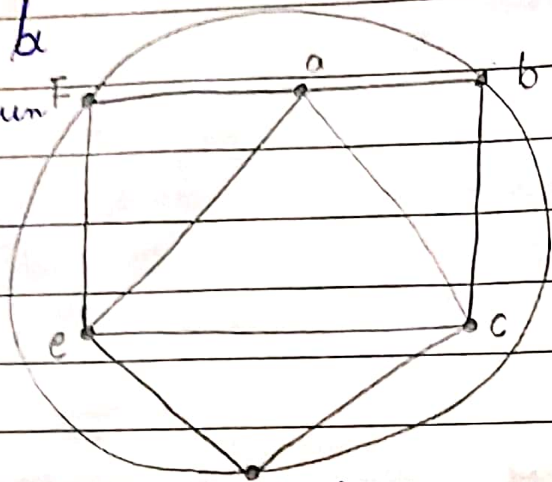
Q Consider the graph as shown in figure
(i) Is It a complete graph ?
(ii) Is It a planar graph ? If so, find the number of regions .



(i) No , It Is not a complete graph as (a , d) , (F , c) , (b , e) Edges are not present In the given graph .
(ii) As , It can be re - drawn In which no

no two Edges will cross each other. So, It Isn't A planar graph.

It can be re-drawn As



∴ We have to find the region of the graph

∴ $\boxed{r = e - v + 2}$

∴ A/q, $e = 12$, $v = 6$

∴ $r = 12 - 6 + 2$

$\boxed{r = 8}$

WALK AND PATH IN A GRAPH =

(i) WALK =

→ It Is A finite Alternating sequence $v_1 e_1 v_2 e_2$ $v_3 e_3$ ---- $v_n e_n$ Of vertices And Edges, ~~begning~~ beginning And Ending with same Or different vertices.

(ii) LENGTH OF A WALK = The number Of Edges In A walk Is called Length of A walk.

Date
Page

(iii) CLOSED AND OPEN WALK = A walk Is said to be closed If It's Origin And terminious vertex ($v_0 = v_m$) Is Equal, Otherwise It Is called As open walk.

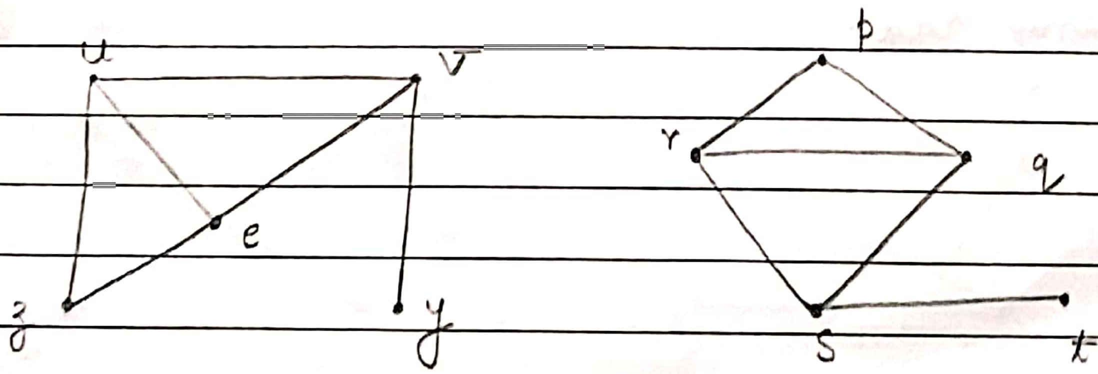
(iv) PATH = A walk Is said to be A path If all the vertices Are not repeated.

ISOMORPHISM OF A GRAPH =

→ Two graph G And G' Are said to be Isomorphic If It satisfies

(i) Number of vertices Are same.
(ii) Number of Edges Are same.
(iii) Degree of Each vertices In both graphs would be same
(iv) Vertex correspondance And Edge correspondance needs to be valid.

Q.



→ $\boxed{n(V) = n(V')}$

→ $\boxed{n(E) = n(E')}$

→ An Equal number of vertices with given

$u = 3$	$p = 2$
$v = 3$	$q = 3$
$x = 3$	$r = 3$
$y = 1$	$s = 3$
$z = 2$	$t = 1$

(iv)

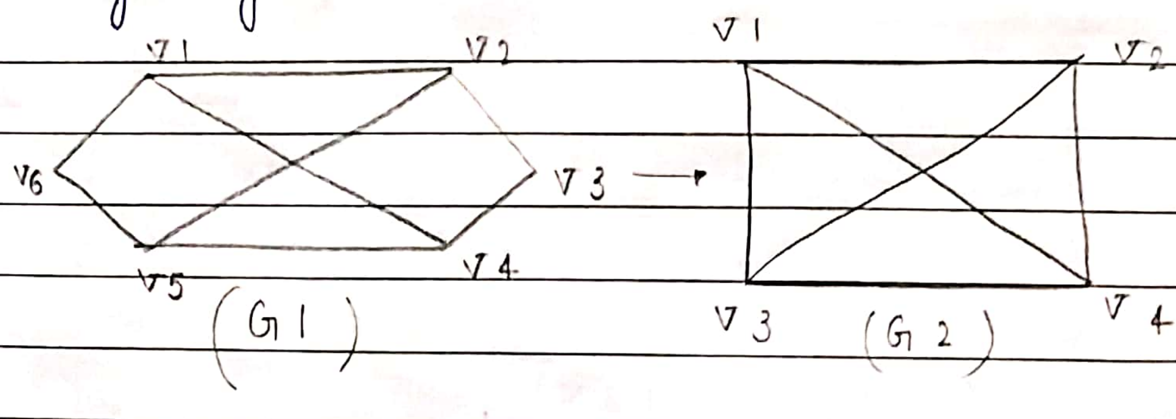
$$\left.\begin{array}{l} y \rightarrow t \\ z \rightarrow p \\ v \rightarrow s \\ u \rightarrow r \\ x \rightarrow q \end{array}\right\} \Longrightarrow \text{Both are Isomorphic.}$$

HOMOMORPHISM OF A GRAPH :

→ Two graph G1 and G2 are said to be a homomorphic graph If G2 can be Obtained from G1 by a sequence of sub-divisions of Edges G1.

In other words, we can Introduce vertices of degree In any edge.

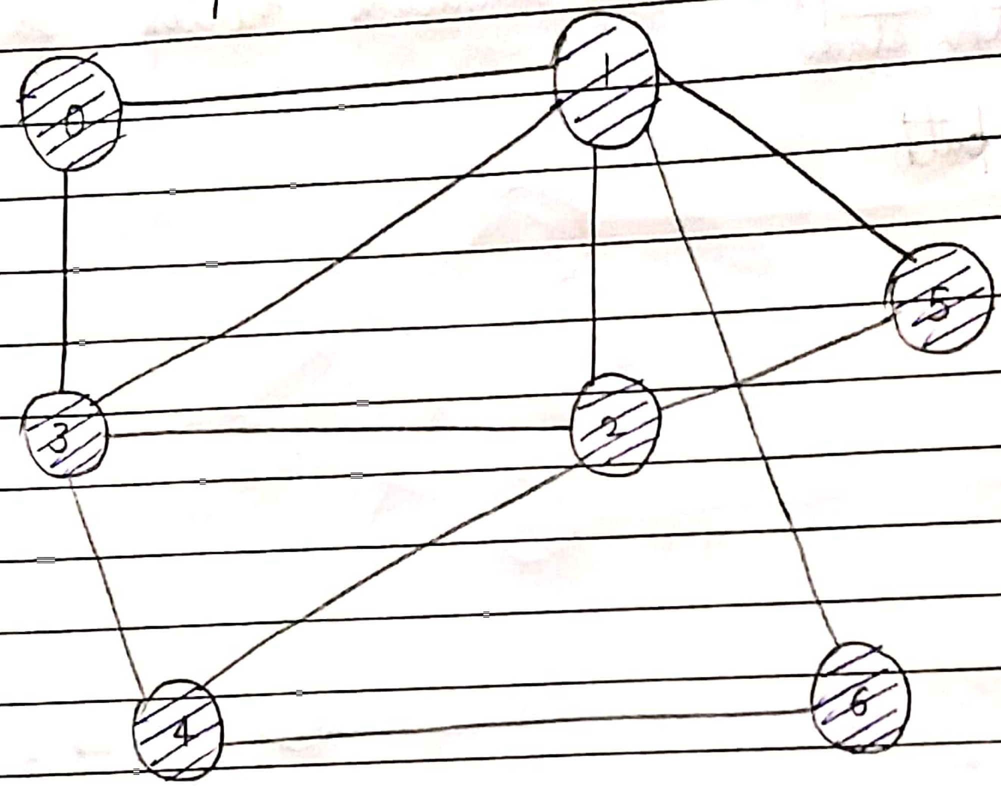
Eg :



GRAPH TRAVERSALS :

1. BFS (level Order) Breadth - first Search
2. DFS . Depth first search

Q. BFS path



Adjacent nodes :

0 → 1, 3

1 → ~~0~~, 3, 2, 5, 6

~~2 → 1, 5, 4~~, 3 → ~~1~~, ~~0~~, 2, 4

2 → ~~1~~, 5, 4, ~~3~~

5 → ~~1~~, ~~2~~

6 → ~~1~~, 4

4 → ~~2~~, ~~3~~, ~~6~~

Queue :

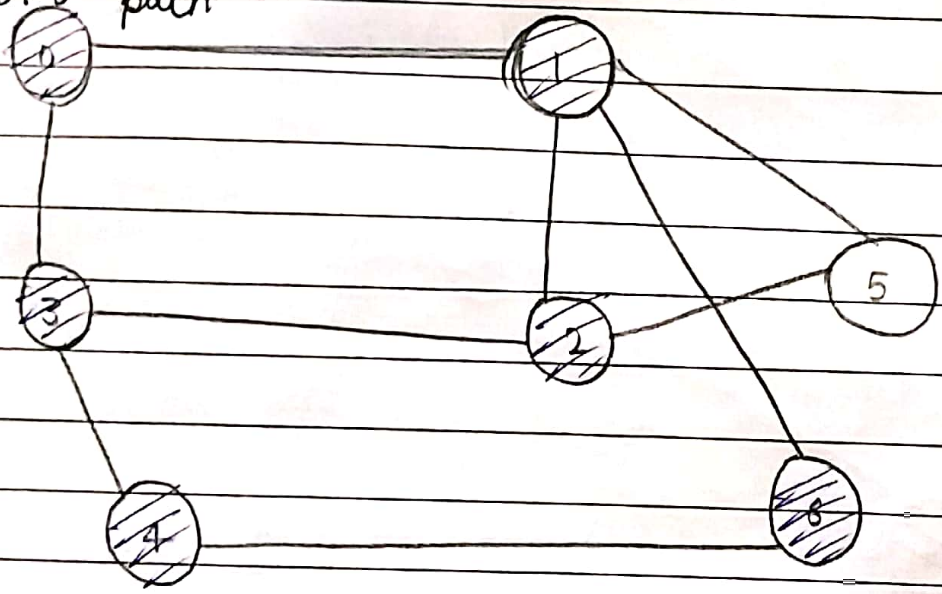
~~0~~	~~1~~	~~3~~	~~2~~	~~5~~	~~6~~	~~4~~

$\boxed{\text{Result} = 0\ 1\ 3\ 2\ 5\ 6\ 4}$

② DFS =

- → Stack Is Used.
- → It works In LIFO manner.
- → We will go deeper till reach the dead End And we have to backtrack.
- → Choose Any One Adjacent vertex And push Into the stack.

Q. DFS path



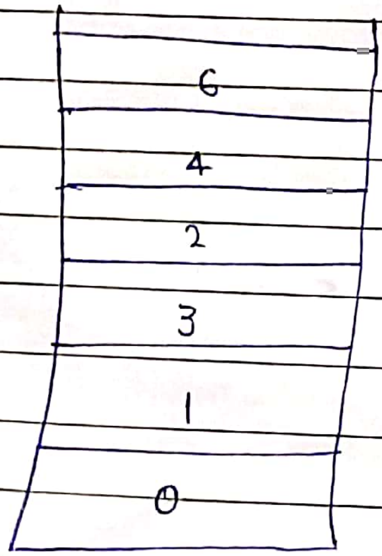
1 → 3, 2, 5, 6

3 → 2, 4

2 → 1, 3, 5, 4

4 → 2, 6

6 → 1, 4



$\boxed{\text{Result} = 0, 1, 3, 2, 4, 6, 5}$

We cannot move further from 6

Date
Page

- → Now we will start backtracking.
- → 6 will be popped from the stack.
- → Now go to 4 but 4 Is not having Any Unvisited vertex. It will be popped Out.
- → Go to 2, It has 5 vertex As Unvisited vertex. Delete 5.
- → Backtrack from 5 to 2, then, 2 will be delete from the stack.
- → Pop Out 3, 1 And 0 Afterwards.
- → Now stack Is Empty And this Is the Indication that we have to stop

PERMUTATIONS AND COMBINATIONS :

(i) **SUM RULE** = Independent $(m+n)$

(ii) **PRODUCT RULE** = Sequences $(m * n)$

Q. In a class there are 10 boys and 8 girls. Teacher has to select either a boy or a girl.

→ ∵ Independent

∴ Number of subsequences = $(m+n)$

∴ A/q,

$$\boxed{(10+8) = 18 \text{ students}}$$

Q. $$\frac{10!}{8!} = \frac{10 \times 9 \times \cancel{8!}}{\cancel{8!}} = 90$$

Note = $$\boxed{{}^nP_r = \frac{n!}{(n-r)!}}$$

Q. What will be the value of =

(i) $$\frac{4 \times n!}{(n-3)!} = \frac{(n+1)!}{(n-2)!}$$

$$\Rightarrow \frac{4 \times \cancel{n!}}{(n-3)!} = \frac{(n+1)\cancel{(n!)}}{(\cancel{n+1}-3)!}$$

$$\Rightarrow \frac{4}{\cancel{(n-3)!}} = \frac{(n+1)}{(n-2)\cancel{(n-3)!}}$$

$$\Rightarrow 4(n-2) = (n+1) \quad \Rightarrow \boxed{n = 3}$$

Date
Page

(ii) $6 \times \frac{n!}{(n-3)!} = 3 \times \frac{(n+1)!}{(n-2)!}$

$\Rightarrow \frac{6 \times \cancel{n!}}{\cancel{(n-3)!}} = \frac{3 \times (n+1)\cancel{(n!)}}{(n-2)\cancel{(n-3)!}}$

$\Rightarrow 6(n-2) = 3(n+1)$

$\Rightarrow 6n - 12 = 3n + 3$

$\Rightarrow 6n - 3n = 3 + 12$

$\Rightarrow \cancel{3}n = \cancel{15}\ 5$

$\boxed{n = 5}$

Q. There are 10 persons called for an Interview. Each Is capable for the job. How many permutations must be required to get selected?

$\Rightarrow {}^{10}P_4$

$\because \boxed{{}^{n}P_{r} = \frac{n!}{(n-r)!}}$

$\therefore$ A/q,

$\Rightarrow {}^{10}P_4 = \frac{10!}{(10-4)!} = \frac{10!}{6!}$

$\Rightarrow \boxed{{}^{10}P_4 = \frac{10 \times 9 \times 8 \times 7 \times \cancel{6!}}{\cancel{6!}} = 5040}$

BOOLEAN ALGEBRA =

→ Discovered by George Bool.
→ Introduced to design digital circuit.

Scanned with CamScanner

Date
Page

SOP (Sum of Product) =

→ Zero As A compliment.

→ A product term which contains All the literals either In complimented Or Uncomplimented form Is called minterm.

→ Minterm

0 0 0	a' b' c'	→	m0
0 0 1	a' b' c	→	m1
0 1 0	a' b c'	→	m2
0 1 1	a' b c	→	m3
1 0 0	a b' c'	→	m4
1 0 1	a b' c	→	m5
1 1 0	a b c'	→	m6
1 1 1	a b c	→	m7

$\therefore F(a, b, c) = (a'b'c' + ab'c + abc' + abc)$

$\boxed{F(a, b, c) = m0 + m5 + m6 + m7)}$

(Hypothetical Scenario)

Compliment $= \sum(0, 1, 2, 3, 4, 5, 6, 7) - \sum(0, 5, 6, 7)$

$\boxed{\text{Compliment} = \sum(1, 2, 3, 4)}$

a.

Scanned with CamScanner

Date

Page

POS (Product of Sum) =

→ In product of sum, One is compliment

~~max~~ Maxterm

0 0 0	a b c	$\longrightarrow$	m0
0 0 1	a b c'	$\longrightarrow$	m1
0 1 0	a b' c	$\longrightarrow$	m2
0 1 1	a b' c'	$\longrightarrow$	m3
1 0 0	a' b c	$\longrightarrow$	m4
1 0 1	a' b c'	$\longrightarrow$	m5
1 1 0	a' b' c	$\longrightarrow$	m6
1 1 1	a' b' c'	$\longrightarrow$	m7

$\because F(a,b,c) = (a+b+c) \cdot (a+b+c') \cdot (a+b'+c)$

$\boxed{F(a,b,c) = m0 \cdot m1 \cdot m2}$

A/q.

Compliment $= \sum(0,1,2,3,4,5,6,7) - \sum(0,1,2)$

$\Rightarrow \boxed{\text{Compliment} = \sum(3,4,5,6,7)}$

Q. Consider the boolean Expression $F(x,y,z) = x(y'z)'$. Reduce It to SOP.

$\boxed{F(x,y,z) = xyz'}$

Scanned with CamScanner

Minterm

0 0 0	$a' \; b' \; c'$	$\longrightarrow$	m_0
0 0 1	$a' \; b' \; c$	$\longrightarrow$	m_1
0 1 0	$a' \; b \; c'$	$\longrightarrow$	m_2
0 1 1	$a' \; b \; c$	$\longrightarrow$	m_3
1 0 0	$a \; b' \; c'$	$\longrightarrow$	m_4
1 0 1	$a \; b' \; c'$	$\longrightarrow$	m_5
1 1 0	$a \; b \; c'$	$\longrightarrow$	m_6
1 1 1	$a \; b \; c$	$\longrightarrow$	m_7

∴ A/q,

$$\boxed{F(x, y, z) = xyz' \approx a\,b\,c' = 110}$$

The