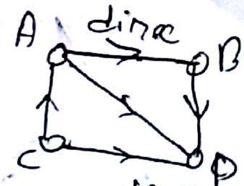
CH-9 GRAPH

$G(V, E)$

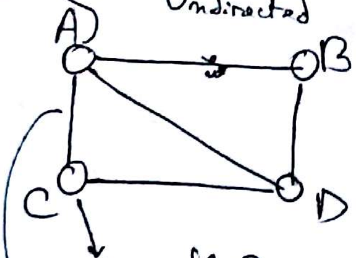
Vertices → Node

Edge → रास्ता



degree of A = 3

Undirected



degree of C = 2

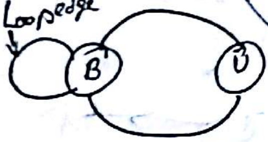
A, B, C, D → Vertices

A ⟶ B
or
A —— B - Edge

Simple graph
→ only one Edge between two vertices

→ Multigraph
If more than 1 edge between 2 vertices

Loop edge

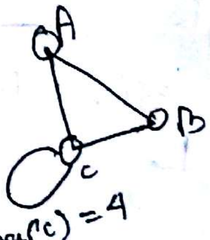


Degree } Undirected

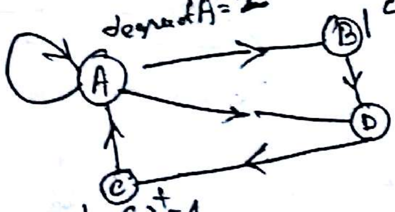
Directed graph { In degree
Out degree

For ~~simple~~ undirected graph

$$\text{No of Edge} = \frac{\text{Add All the degree}}{2} = \frac{3+3+2+2}{2} = \frac{10}{2} = 5$$



$deg(C) = 4$



degree of $A^{+} = 2$

$degree(B)^{+} = 1$

$deg(D)^{+} = 2$

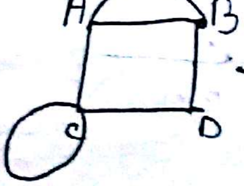
$deg(C)^{+} = 1$

$+$ = indegree

No of edge = sum of degree $(+)$

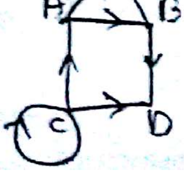
Graph Representation

- Adjacency list
- Adjacency Matrix
- Incidence list



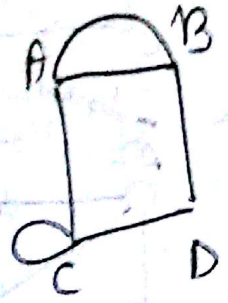
→

	Adjacency list
A	B, C, B
B	A, D, A
C	A, D, C
D	B, C

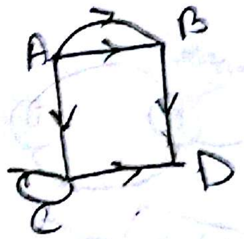


	Adj List
A	B
B	D
C	C, A, D
D	

Adjacency Matrix

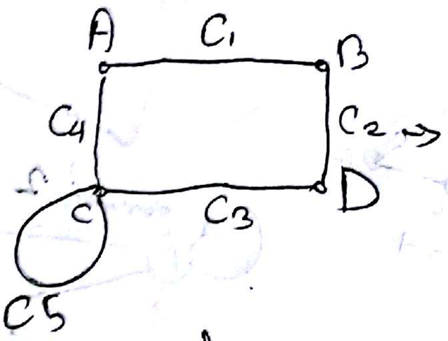


	A	B	C	D
A	0	2	1	0
B	2	0	0	1
C	1	0	1	1
D	0	1	1	0

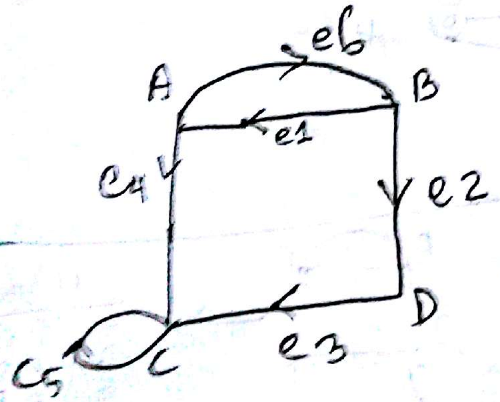


	A	B	C	D
A	0	2	1	0
B	0	0	0	1
C	0	0	1	1
D	0	0	0	0

Incidence List



	c_1	c_2	c_3	c_4	c_5
A	1	0	0	1	0
B	1	1	0	0	0
C	0	0	1	1	1
D	0	1	1	0	0



Let out → 1
in → −1

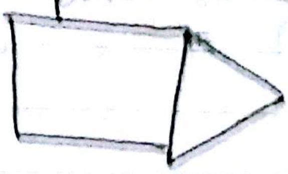
	e_1	e_2	e_3	e_4	e_5	e_6
A	−1	0	0	1	0	1
B	1	1	0	0	0	−1
C	0	0	−1	−1	1/−1	0
D	0	−1	1	0	0	0

Connected graph → edge

Date:

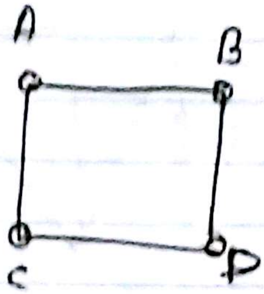
Sat Sun Mon Tue Wed Thu Fri

Euler path → No edge is repeated

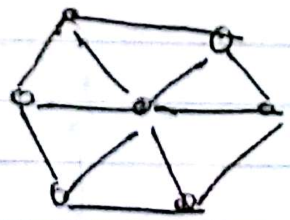


Euler circuit → Visit each edge once and come back to start.

Hamilton Circuit → vertex



each vertex visited 1 time and not



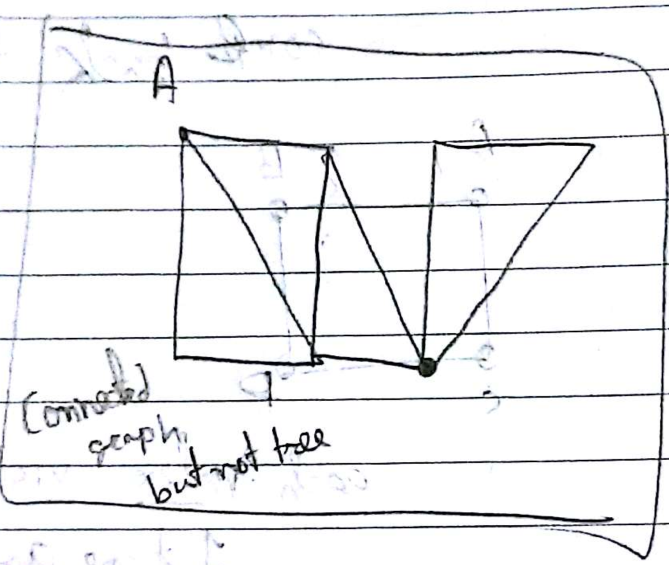
Not Euler Circuit

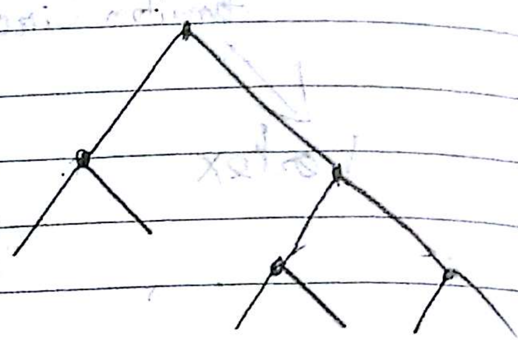
but Hamilton circuit

and ...

CH-10

Trees





Tree

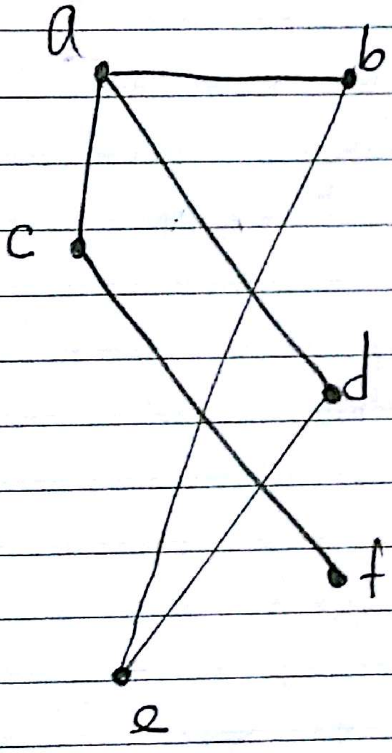
G_2 a G_1

N 2and9 [illegible] Mgt 211

Total mark - 40

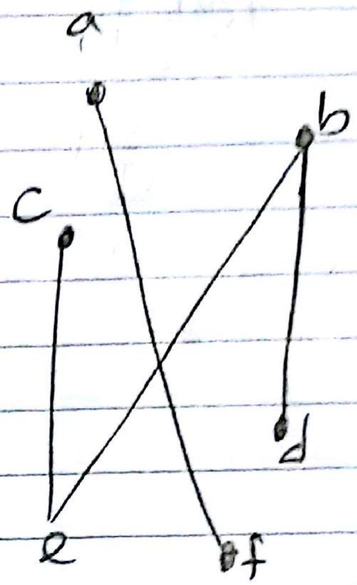
Case :

G3



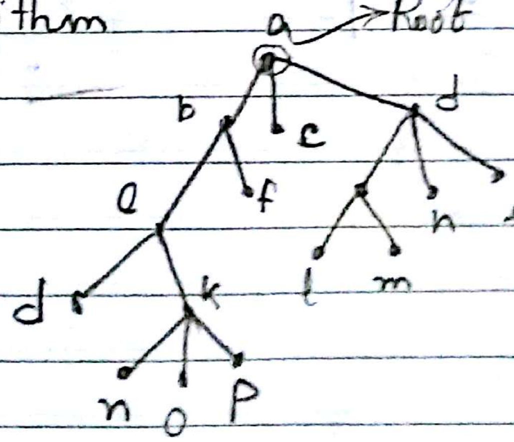
abcd has cycle so not a tree

G4



disconnected so not a tree

Traversal Algorithm



s-l-r Pre-order → visit root then visit left to right

Inorder ⇒ l-s-r

Post order

ze a b e j k n o p f c d g l m h i

In: j e k n k o p b f a c

l g m d h i

Post → ~~R self~~

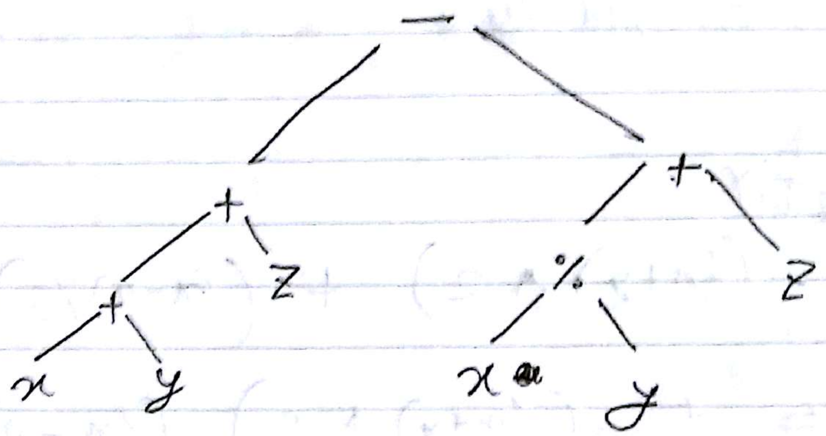
Left Right root

~~root~~

→ self

Rooted tree

$$\{(x+y)+z\} - \{(x \% y)+z\}$$



Infix, prefix, postfix

Date :

Sat Sun Mon Tue Wed Thu Fri

$\sim x/(y/z)$

$\rightarrow = x/(/yz)$

$= /x/yz$

Subject: Date: Sat Sun Mon Tue Wed Thu Fri

$((x+y)\uparrow 2) + ((x-4)/3)$

$= (((x+y)2\uparrow) \quad ((x-4)/3)+$

$= (x+y)2\uparrow \quad (x-4)3/+$

$= x\,y+2\uparrow \quad x\,4-3/+$

Prefix

$+ - \underbrace{* \; 2\,3}\,5 / \underbrace{\uparrow 2\,3}\,4$

$= + - 6\,5 / 8\,4$

$\cancel{= + 1 / 8\,4}$

$= + 1 / 8\,4$

$= + 1\;2$

$= 3$

Post fix

$7\,\underbrace{2\,3*} - 4 \uparrow \underbrace{9\,3/} +$

$= \underbrace{7\,6\,-}\,4\uparrow 3 +$

$= \underbrace{1\;4\uparrow}\,3+$

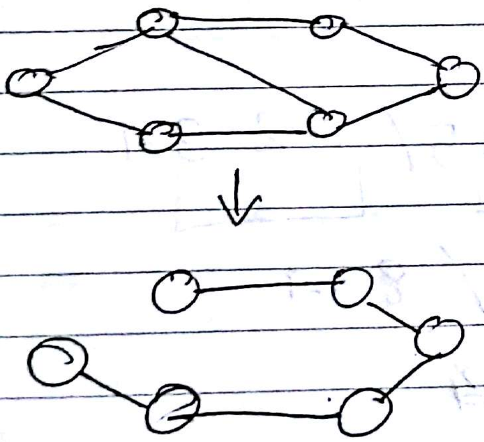
$= 1\;3+$

$= 4$

Papers King

- Spanning tree
- Minimum Cost Spanning Tree

Graph to tree convert



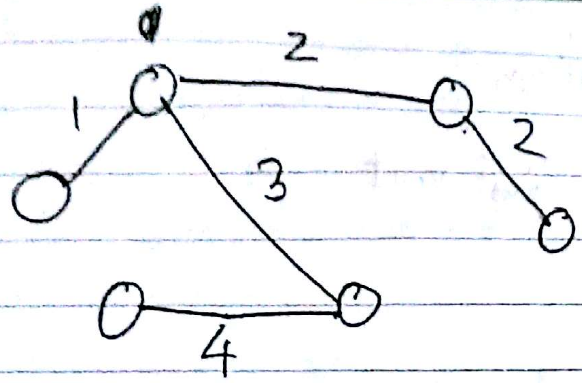
spanning Tree

a graph can have multiple spanning tree

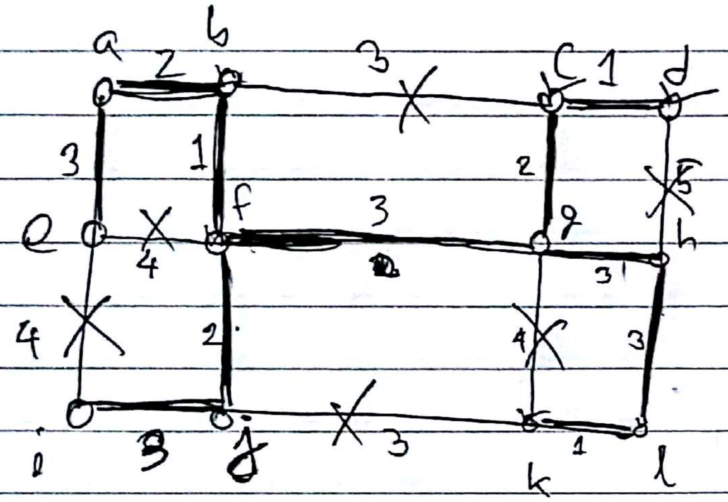
Subject:

Date:

Sat Sun Mon Tue Wed Thu Fri



min cost spanning tree



Kruscal Algo

* go to weight least the start then gradually make sets

1

Merge

~~{c, d, g}~~

{f, b, a, j, e, i, c, d, g}, h, k, l

Merge

~~{k, l}~~

sum of weights = minimum cost

can have differt type.

but cost is one.

Subject :

Date :

Sat Sun Mon Tue Wed Thu Fri

circut → has to return to starting node after visiting all nodes

path → dont return to start node just visit all the Nodes