* Permutation :-

(i) TYPE I : No repitition

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

(ii) TYPE II : when repititon

$$^{n}P_{r} = \frac{n!}{r_1! \times r_2! \times r_3! \ldots}$$

(iii) TYPE III :

$$^{n}P_{r} = n^{r}$$

* <u>combination</u> :-

$$^{n}C_{r} = \frac{n!}{r!(n-r)!}$$

Rainbow

PAGE:
DATE: / /

Discrete Math

Unit 4

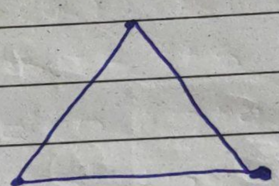
Graph solved Question

NOV_DEC_2022

Q3)

a) - A simple graph is unweighted, undirected graph with no self loops and parallel edges.

~~A simple graph is said to be opposite of complete graph.~~



- It is not possible to draw a simple graph, with 4 vertices and 7 edges.
- Becouse in simple graph, each edge connect two distinct vertices, and having 7 edges with 4 vertices will violate the rule that each vertex can have maximum degree of 3. This violates the handshaking lemma, which state sum of degree of all vertices is twice the no of edges.

- In this case there is 4 vertices to find maximum edges there is formula

$$\frac{n \cdot (n-1)}{2} = \frac{4(4-1)}{2}$$

$$= \frac{12}{2}$$

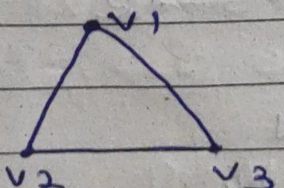
$$= 6$$

∴ 6 maximum edges can be there with four vertices.

5)

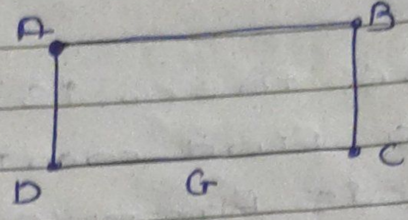
(i) <u>complete</u> <u>graph</u> :-

- A graph G is called complete graph if every vertex in G is connected with every other vertex.



(ii) <u>Regular Graph</u> :-

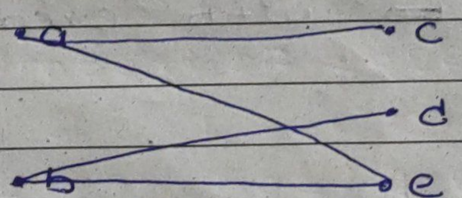
A Graph G is said to be regular graph if degree of each vertex is same.



$d(A) = 2$
$d(B) = 2$
$d(C) = 2$
$d(D) = 2$

iii) Bipartite graph:-

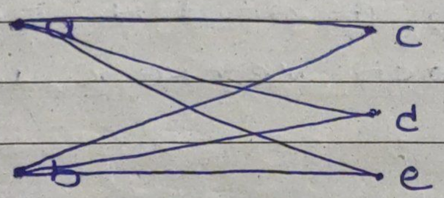
A Graph (G) is said to be Bipartite graph if its vertex set can be Portition into two set as V_1 and V_2 such that no vertex in same Portition can be adjacent.



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

iv) complete Bipartite graph:-
~~It is gra~~
A graph G is said to complete Bipartite graph whose vertex set can be partition

into two subset of m and n each vertex of subset m is uniquely connect with each vertex of subset n.

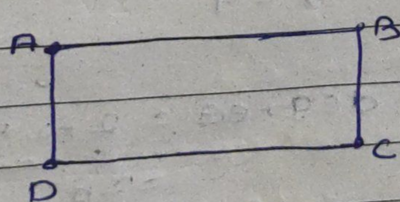


$m = \{a, b\}$
$N = \{c, d, e\}$

v) Path and Circuit :-

- In graph a path is a sequence of vertices where consecutive vertices are covered by edges is called path.

- A circuit is a closed path, it start and ends with same vertex.



Path :- A - B - C - D
Circuit :- A - B - C - D - A

Rainbow

PAGE:
DATE: / /

Q3 c)

(1) In Graph G and H both are having same no of vertices and edges 7 vertices and 7 edges.

(2) In graph G and H both are having each vertex with degree 2

(3) In graph G and H both are having minimum circuit as 7.

(4) So it is isomorphic.

Q4)
b)

(i) Number of edges :-

- graph has 9 vertices of degree 2, 2, 2, 3, 3, 4, 4, 5

$V = 9$

$\therefore$ Total degree $= 2+2+2+3+3+4+4+5$
$= 28$

$e = \frac{\text{Total degree}}{2} = \frac{28}{2} = 14$ edge

(iii) No of faces :-

Euler formula

$$F = e - v + 2$$
$$= 14 - 9 + 2$$
$$= 5 + 2$$
$$= 7$$

$\boxed{\text{No of faces} = 7}$

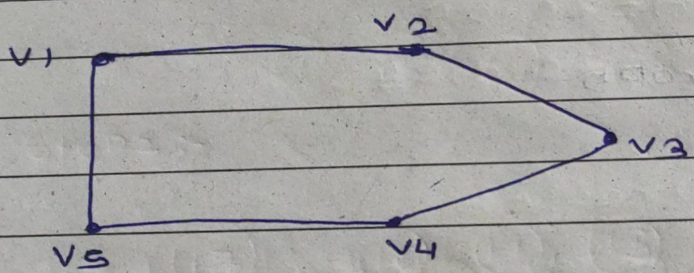
c) Explain following statement with example.

"Every graph

May_June 2023

Q3)

b) In discreate mathematics, an adjacency matrix represent connection between vertices in graph. It's a square matrix where entry A[i][j] is 1 if there is edge between vertices i and j, and 0 otherwise.

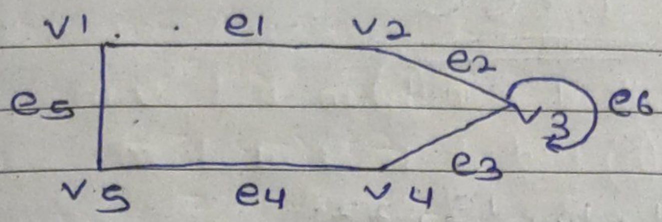


$$
\begin{array}{c}
\\ v1 \\ v2 \\ v3 \\ v4 \\ v5
\end{array}
\begin{array}{c}
\begin{array}{ccccc} v1 & v2 & v3 & v4 & v5 \end{array} \\
\begin{bmatrix}
0 & 1 & 0 & 0 & 1 \\
1 & 0 & 1 & 0 & 0 \\
0 & 1 & 0 & 1 & 0 \\
0 & 0 & 1 & 0 & 1 \\
1 & 0 & 0 & 1 & 0
\end{bmatrix}
\end{array}
$$

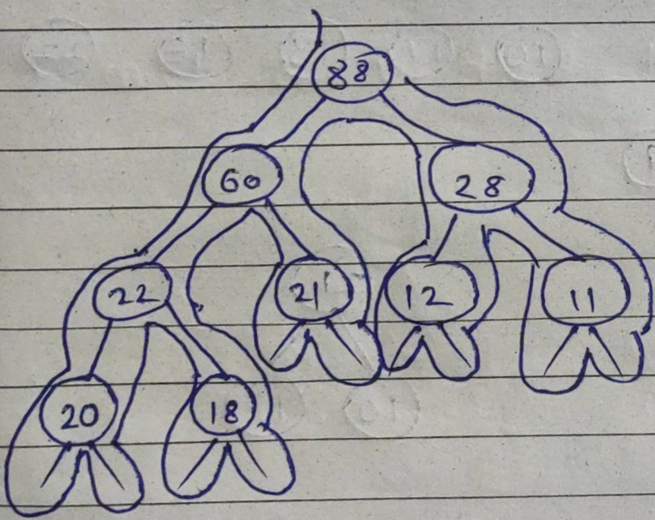
· Incidence matrix :-

an incidence matrix describes relationship between vertices and edges.
Entry m[i][j] is 1 if vertex is incident, 2 if it is self loop, 0 otherwise.

46 6 0 .



$$\begin{array}{c} \\ v1 \\ v2 \\ v3 \\ v4 \\ v5 \end{array}
\begin{array}{c} \begin{array}{cccccc} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 \end{array} \\
\begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 2 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \end{bmatrix} \end{array}$$



1 Preorder - 88, 60, 22, 20, 18, 21, 28, 12, 11,
2 Inorder - 20, 22, 18, 60, 21, 88, 12, 28, 11
3 postorder - 20, 18, 22, 21, 60, 12, 11, 28, 88

* Expand using binomial theorem

TYPE - 1

① $(x^4 + 2)^3$
a b

$n = 3$
$a = x^4$
$b = 2$

$$\boxed{t_{r+1} = {}^nC_r \times a^{n-r} \times b^r}$$ * imp

when
1) $r = 0$

$\cancel{{}^3C_0 \times x^3 \times x^3}$
${}^3C_0 \times x^3 \times 1$
$1 \times x^3 \times 1$
$\underline{x^3}$

when $r = 1$
② ${}^nC_r \times a^{n-r} \times b^r$

${}^3C_1 \times x^2 \times 2^1$
$\frac{3!}{1!(2!)} \times x^2 \times 2$
$3x^2 \times 2 \Rightarrow 6x^2$

③ when $x = 2$
${}^nC_r \times a^{n-r} \times b^r$
${}^3C_2 \times x^1 \times 2^2$
$3x \times 4$
$12x$

⑤ when $x = 3$
${}^nC_r \times a^{n-r} \times b^r$ | ${}^3C_3 \times x^0 \times 2^3$
$\frac{3!}{3!(0!)} 1 \times 1 \times 8 = 8$

* Additing all condition.

$\Rightarrow \boxed{x^3 + 6x^2 + 12x + 8}$

$(x^4 - 2)^3$
↓
then we will
$-, +, -, +$

* Find 8th term in the expansion of $(x+y)^{13}$

$r = \text{term} - 1$

$= 8 - 1$

$\boxed{r = 7}$

$a = x$

$b = y$

$n = 13$

$$T_{r+1} = {}^{n}C_{r} \times a^{n-r} \times b^{r}$$
$$= {}^{13}C_{7} \times x^{6} \times y^{7}$$
$$= \frac{13!}{7!(6!)} \times x^{6}y^{7}$$
$$= 1716\, x^{6}y^{7}$$

* $(1+x)^{6}$, x^{3}

$a = 1$

$b = x$

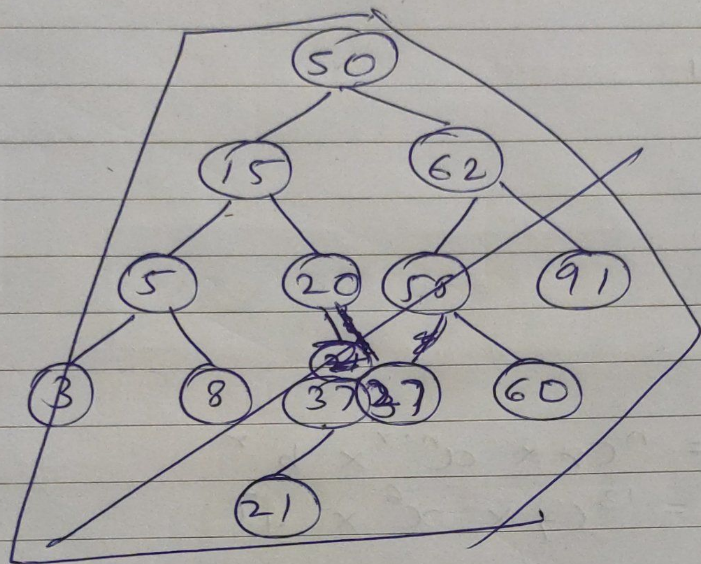
$n = 6$

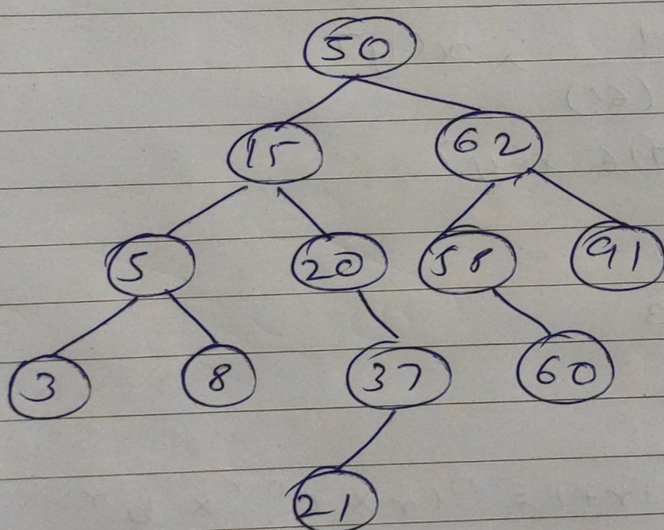
$r = 3$

$$T_{r+1} = {}^{n}C_{r} \times a^{n-r} \times b^{r}$$
$$= {}^{6}C_{3} \times 1^{3} \times x^{3}$$
$$= \frac{6!}{3!(3!)} \times x^{3}$$
$$= \frac{\cancel{6} \times 5 \times \cancel{4}\,2}{\cancel{3} \times \cancel{2}}$$
$$= \underline{\underline{20x^{3}}}$$

* Binary search tree

50, 15, 62, 5, 20, 58, 91, 3, 8, 37, 60, 21

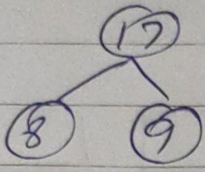




* Huffman coding & prefix code

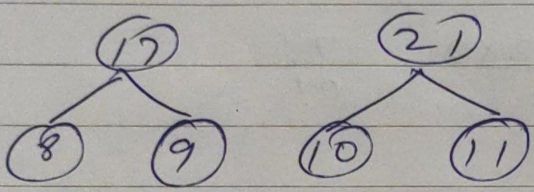
* 8 9 10 11 13 15 22

1



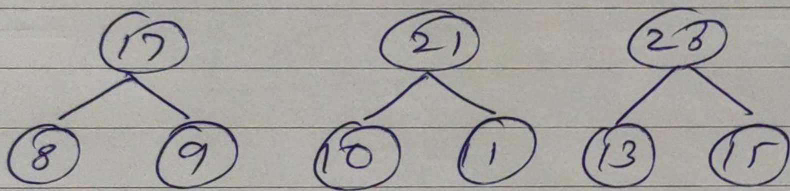
10 11 13 15 22

2



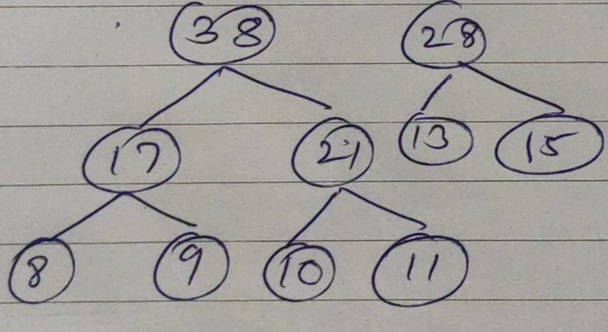
13 15 22

3



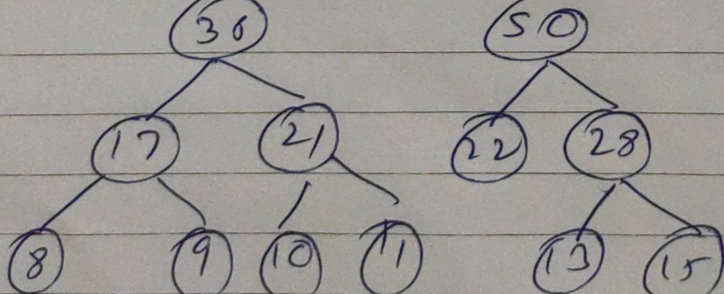
22

4

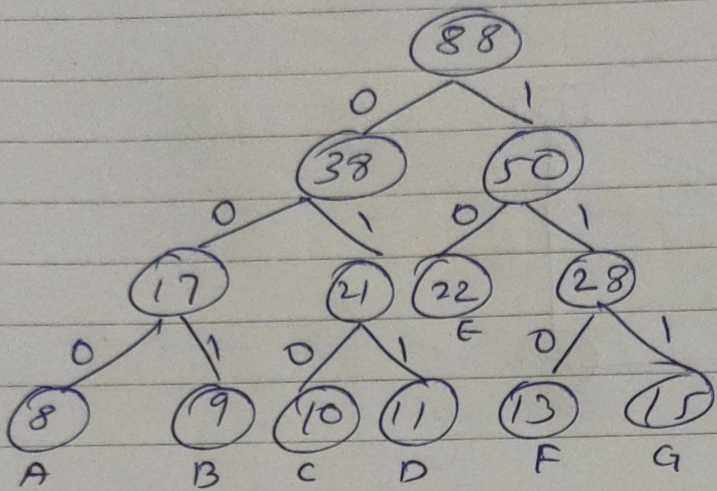


22

5



⑥



8 ⇒ 000

9 ⇒ 001

10 ⇒ 010

11 ⇒ 011

22 ⇒ 10

13 ~~15~~ ⇒ 110

15 ⇒ 111

W(A) = 8 × 3 = 24

W(B) = 9 × 3 = 27

W(C) = 10 × 3 = 30

W(D) = 11 × 3 = 33

W(E) = 22 × 2 = 44

W(F) = 13 × 3 = 39

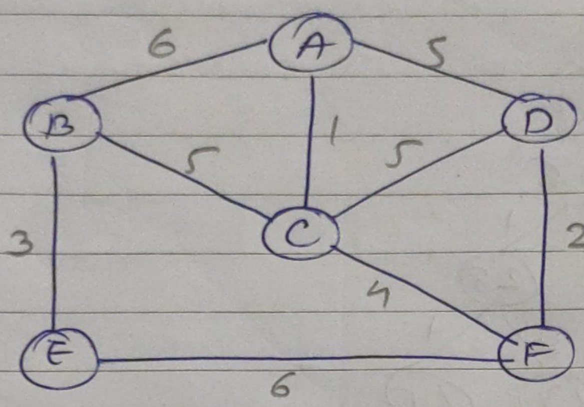
W(15) = 15 × 3 = 45

Total weight ⇒ 242
7(W)

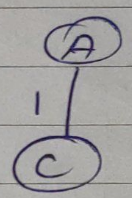
minimum spanning tree.

* Prim's Algorithm

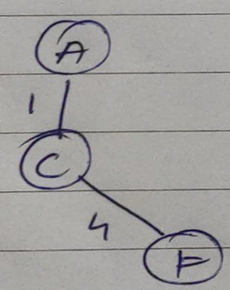
1



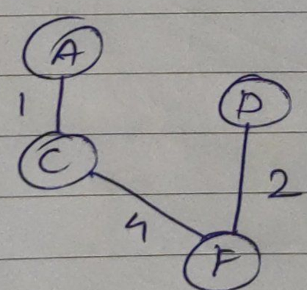
Step 1 :



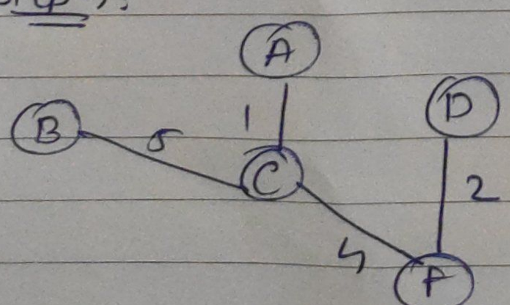
Step 2 :



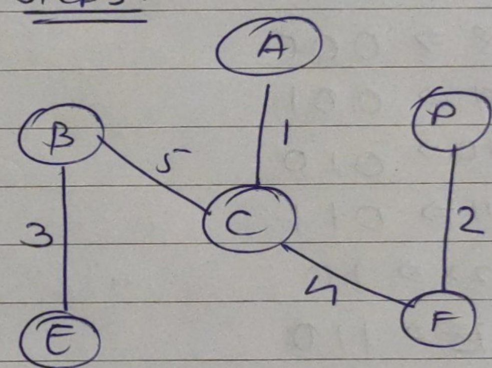
Step 3 :



Step 4 :



Step 5 :



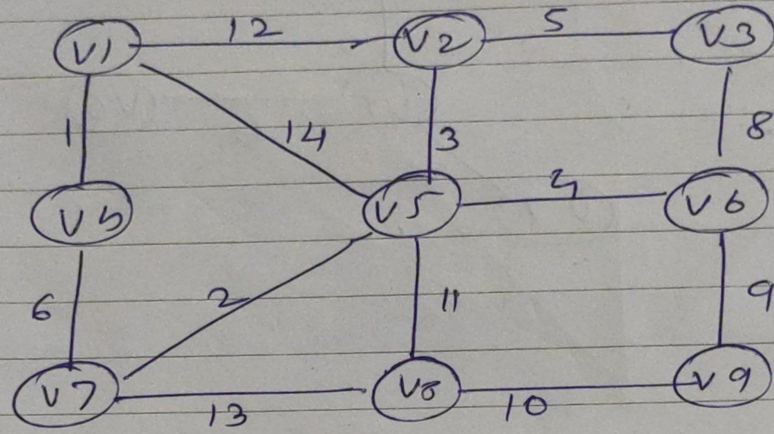
$$W(T) = w(AC) + w(CF) + w(FD) + w(BC) + w(BE)$$

$$= 3 + 5 + 1 + 4 + 2$$

$$= \cancel{20}15$$

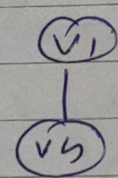
Kruskal Algorithm

①

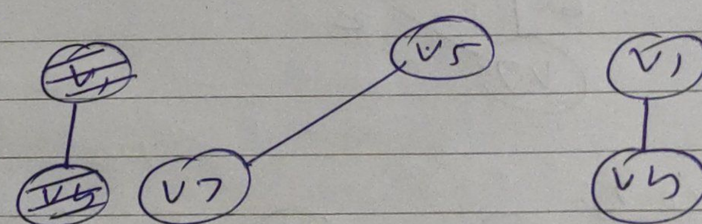


$(V_1 - V_4) = 1$
$(V_5 - V_7) = 2$
$(V_2 - V_5) = 3$
$(V_5 - V_6) = 4$
$(V_2 - V_3) = 5$
$(V_4 - V_7) = 6$
$(V_3 - V_6) = 8$
$(V_6 - V_9) = 9$
$(V_8 - V_9) = 10$
$(V_5 - V_8) = 11$
$(V_1 - V_2) = 12$
$(V_7 - V_8) = 13$
$(V_1 - V_5) = 14$

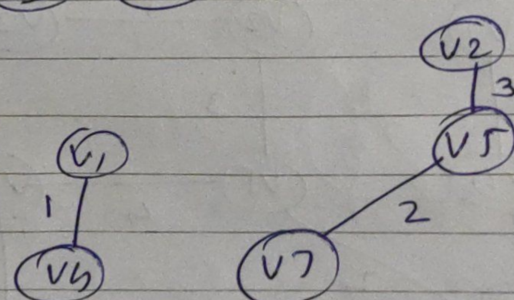
Step 1:



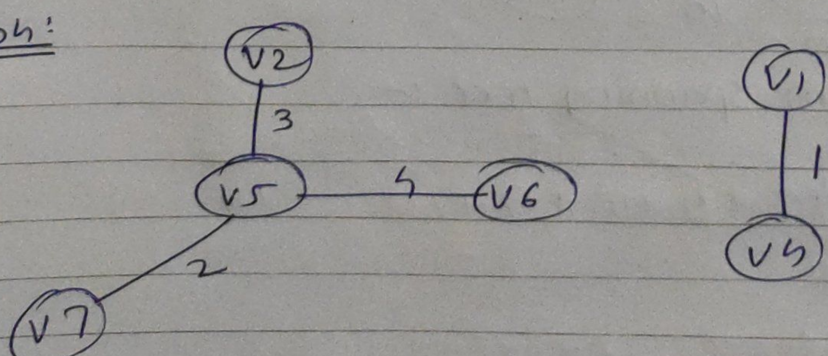
Step 2:



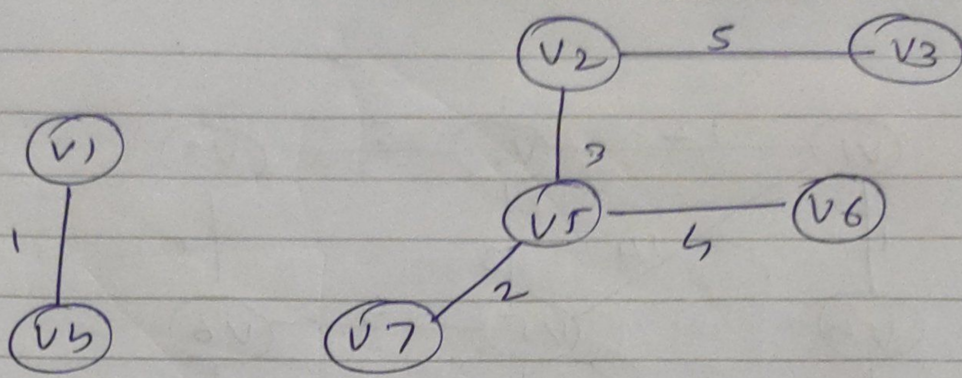
Step 3:



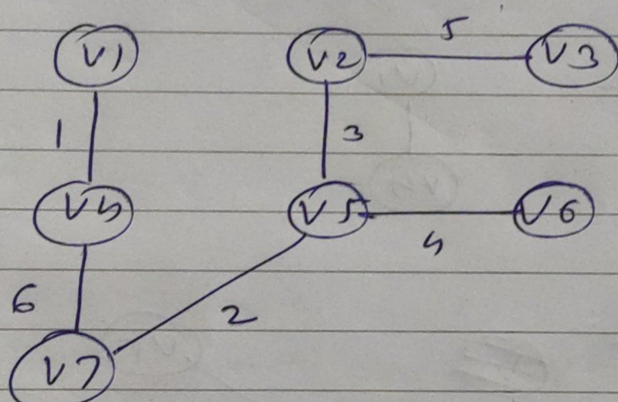
Step 4:



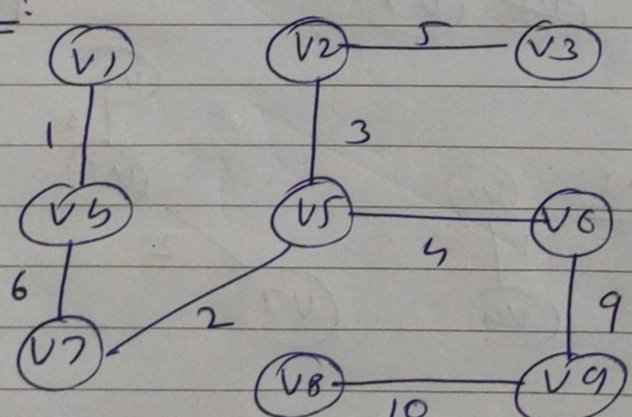
Step 5:



Step 6:



Step 7:



cost of minimum spanning tree :-

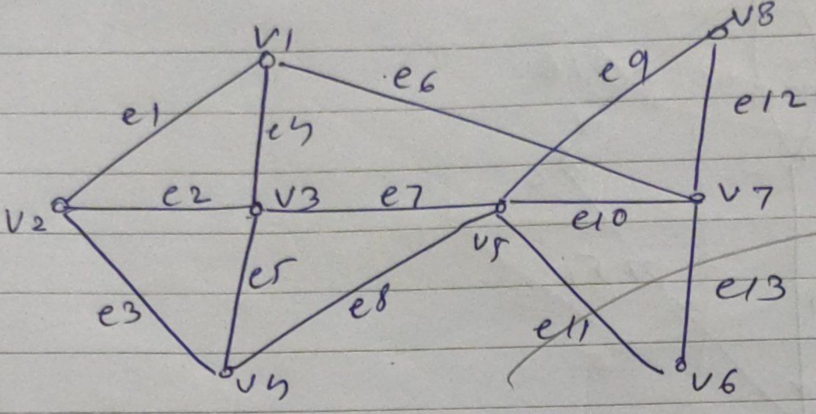
= w(V1, V4) . . . . . .

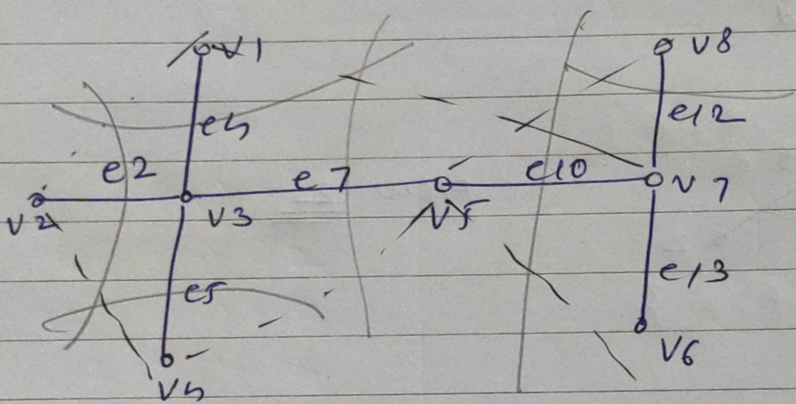
= 1 + 6 + 2 + 3 + 4 + 5 + 9 + 10

= 40

* Fundamental cutset

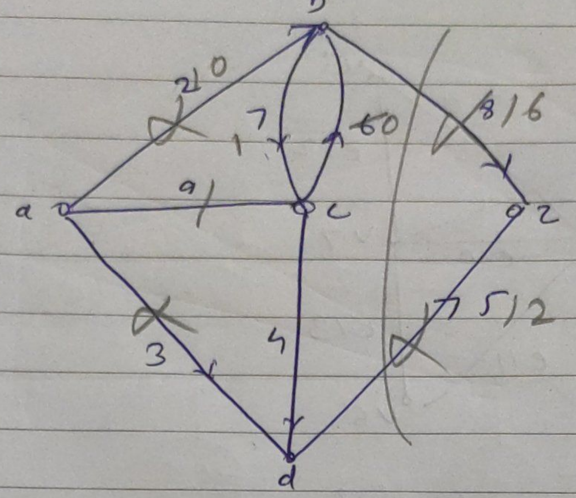
1)



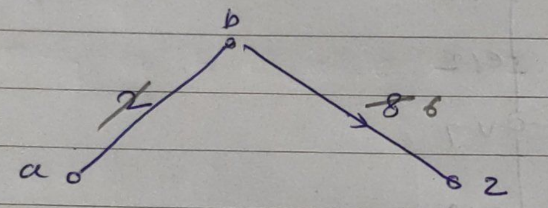


Branch	fundamental cutset
e2	e2, e1, e3
e4	e1, e4, e6
e5	e3, e5, e8
e7	e6, e7, e8
e10	e9, e6, e10, e11
e12	e9, e12
e13	e11, e13

* maximum flow & minimum cut

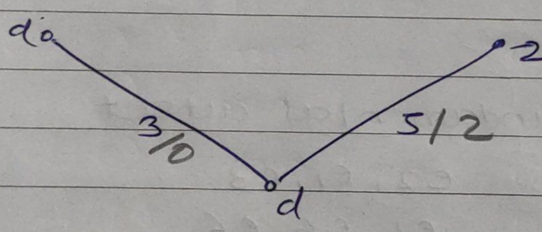


(1)



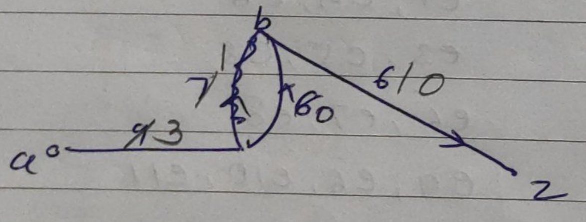
Flow = 2

(2)



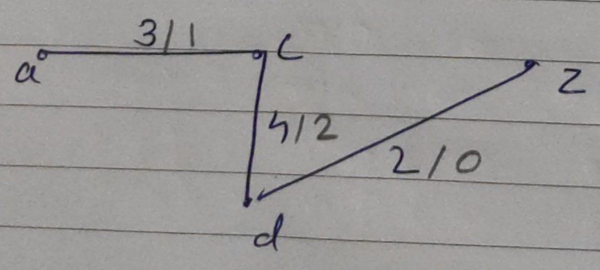
Flow = 3

(3)



Flow = 6

(3)



flow = 2

~~max flow~~ $2+3+6+2 = \underline{13}$

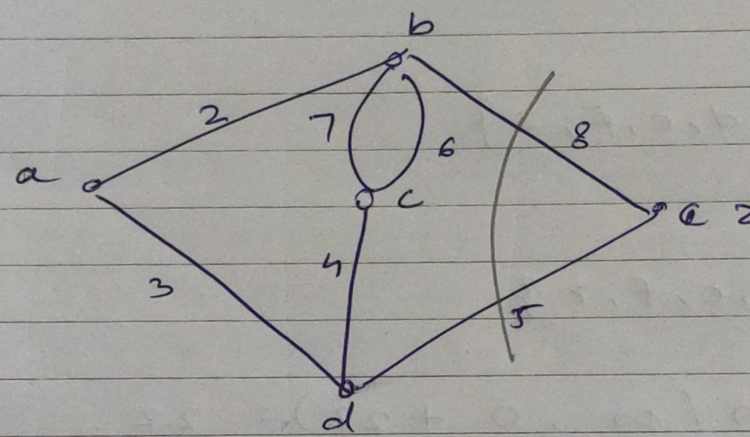
minimum cut $= (b \to z) + dz$

$= 8 + 5$

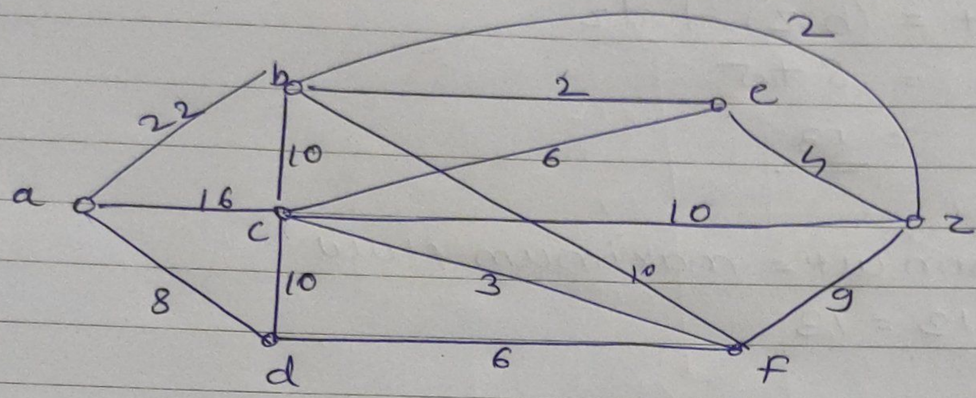
$= \underline{\underline{13}}$

$\therefore$ minimum cut = maximum flow

$13 = 13$



Dijkstra algorithm:



(i) $P = \phi$

$T = \{a, b, c, d, e, f, z\}$

(ii) $P = \{a\}$

$T = \{b, c, d, e, f, z\}$

$l(b) = \min(\infty, 0 + 22) = 22$

$l(c) = \min(\infty, 0 + 16) = 16$

$l(d) = \min(\infty, 0 + 8) = 8$ – min

$l(e) = \min(\infty, 0 + \infty) = \infty$

$l(f) = \min(\infty, 0 + \infty) = \infty$

$l(z) = \min(\infty, 0 + \infty) = \infty$

(iii) $P = \{a, d\}$

$T = \{b, c, e, f, z\}$

$l(b) = \min(22, 8 + \infty) = 22$

$l(c) = \min(16, 8 + 10) = 16$

$l(e) = \min(\infty, 8 + \infty) = \infty$

$l(f) = \min(\infty, 8 + 6) = 14$ – min

$l(z) = \min(\infty, 8 + \infty) = \infty$

(iv) $P = \{a, d, f\}$
$T = \{b, c, e, z\}$

$l(b) = \min(22, 14 + 10) = 22$
$l(c) = \min(16, 14 + 3) = 16$ — min
$l(e) = \min(\infty, 14 + \infty) = \infty$
$l(z) = \min(\infty, 14 + 9) = 23$

(v) $P = \{a, d, f, c\}$
$T = \{b, e, z\}$

$l(b) = \min(22, 16 + 10) = 22$ — min
$l(e) = \min(\infty, 16 + 6) = 22$
$l(z) = \min(23, 16 + 10) = 23$

(vi) $P = \{a, d, f, c, b\}$
$T = \{e, z\}$

$l(e) = \min(22, 22 + 2) = 22$ — min
$l(z) = \min(23, 22 + \infty) = 23$

(vii) $P = \{a, d, f, c, b, e\}$
$T = \{z\}$

$l(z) = \min(23, 22 + 4) = \underline{\underline{23}}$ — min

(viii)
Permanent label of $z = 23$
Hence shortest path length,
from a to z is 23 & shortest path is a - d - f - z

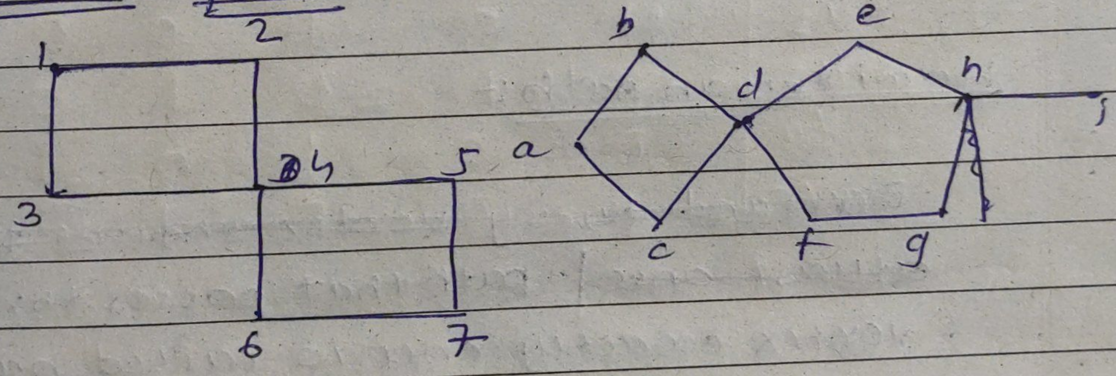
(i) In Graph G and H both are having 7 vertices and 7 edges

(ii) Both graph G and H having each vertex of degree 2.

(iii) In Graph G and H both are having minimum length of circuit as 7.

(iv) Therefore graph G and H are isomorphic to each other.

* Eulerian path



- A graph is Euler path if and only if there are at most two {0 or 2} vertices with odd degree.

- Graph G_1 has zero vertices with odd degree and G_2 has two vertices h and i with odd degree.

  ∴ both G_1 and G_2 are Euler path.

Eulier circuit:

* A graph is Euler circuit if and only if degree of every vertex is even

∴ Graph G_1 and G_2 both are not Eulier circuit

~~Hamiltonian path:-~~

~~Non~~

~~sum of degree of adjacent pair should be greater than vertices minus one then it is Hamiltonian path.~~

Hamiltonian path:-

~~Every and every vertex should be covered atleast once~~ | path that passes through every vertex exactly once is called hamitonian path.

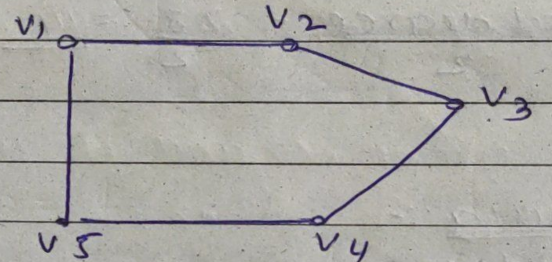
Hamiltonian circuit:-

In graph G_1 we should start and end with same vertex by visiting every vertex exactly at once, then it is Hamiltonian ciruit.

* Adjacency matrix :-

- Adjacency matrix represents connections between vertices in a graph. It's a square matrix where entry $A[i][j]$ is 1, if there is an edge between vertices i and j, and 0 otherwise.

eg.

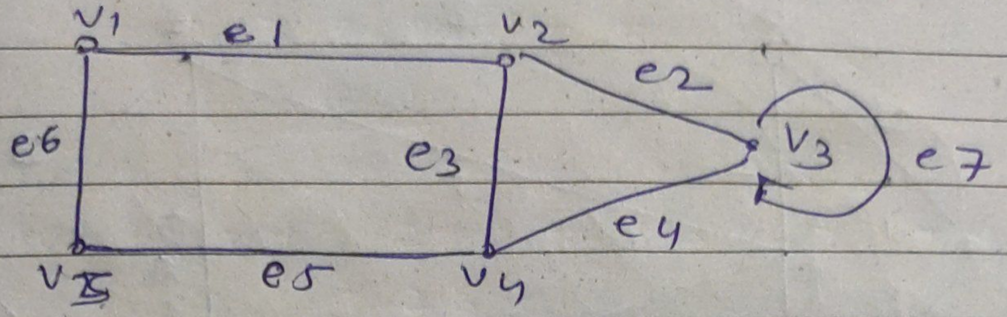


$$\begin{array}{c|ccccc} & V_1 & V_2 & V_3 & V_4 & V_5 \\ \hline V_1 & 0 & 1 & 0 & 0 & 1 \\ V_2 & 1 & 0 & 1 & 0 & 0 \\ V_3 & 0 & 1 & 0 & 1 & 0 \\ V_4 & 0 & 0 & 1 & 0 & 1 \\ V_5 & 1 & 0 & 0 & 1 & 0 \end{array}$$

* Incidence Matrix :-

- An incidence matrix describes relationship between vertices and edges.
- Its a rectangular matrix with row corresponding to vertices and column to edges. Entry $M[i][j]$ is 1 if vertex i is incident to edge j, and 0 otherwise and 2 if it has self loop.

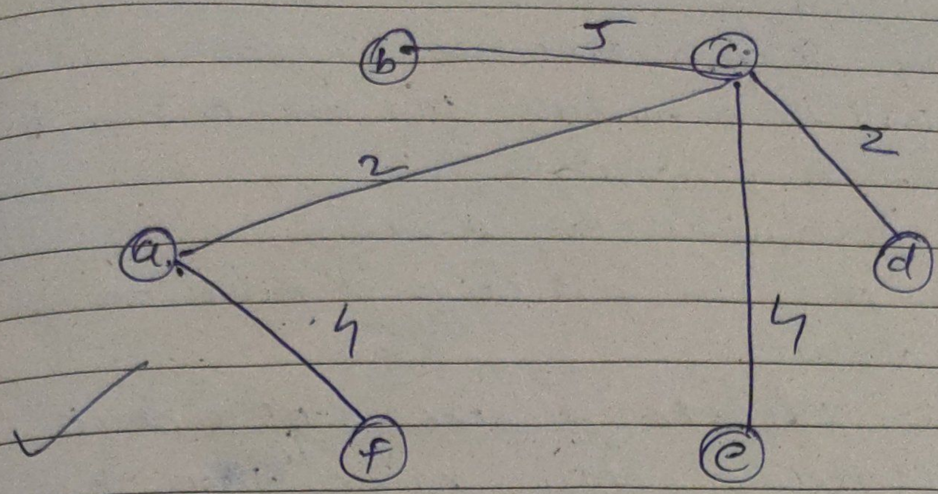
eg-



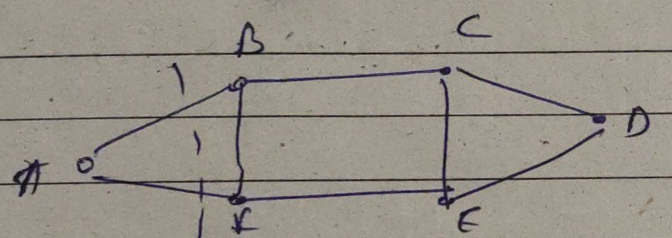
$v_1, v_2 \ldots v_5 \Rightarrow$ vertices

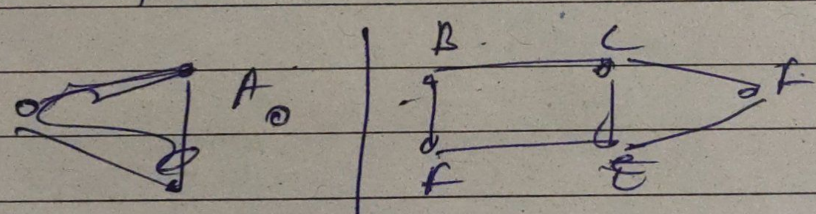
$e_1, e_2 \ldots e_7 \Rightarrow$ edges.

$$\begin{array}{c|ccccccc} & e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & e_7 \\ \hline v_1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ v_2 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ v_3 & 0 & 1 & 0 & 1 & 0 & 0 & 2 \\ v_4 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ v_5 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \end{array}$$

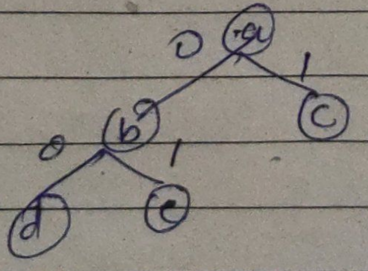


* A cutset is set of nodes in graph whose removal disconnect remaining nodes





* Prefix code :-
  sequence of bit assign to characters in binary tree assign 0 bit to left sub tree and one bit to right sub tree.



d ⇒ 00
e = 01
c = 1

Rainbow
PAGE:
DATE: / /

* Binary Search tree :-
BST is a binary tree which has the following properties.
(i) Each node has a value
(ii) A total order is defined on these values
(iii) The left subtree of a node contains only values less than the node values
(iv) The right subtree of a node contains only values greater than or equal to the nodes values.

Transversal

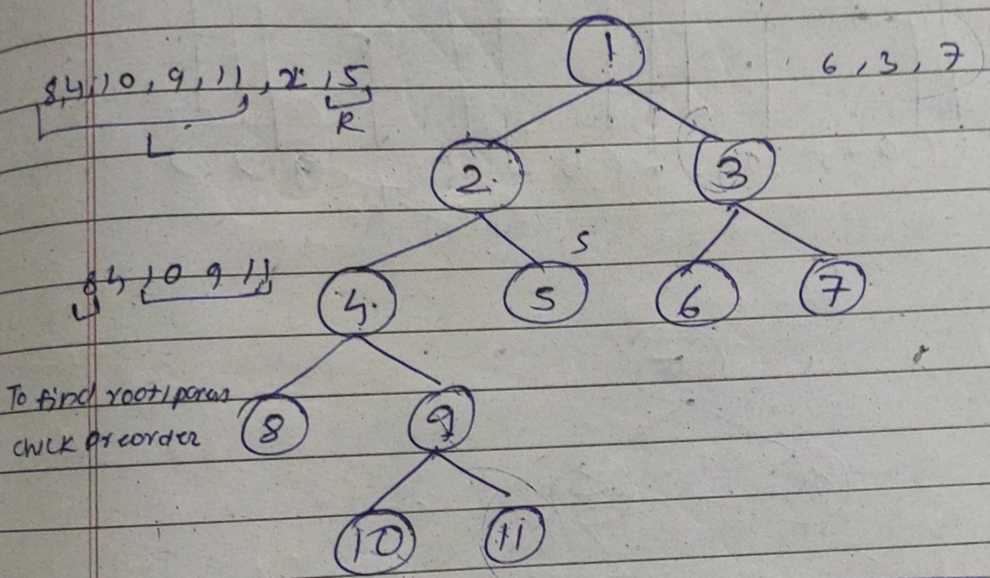
Binary tree

scan left to right

Preorder :- 1 2 4 8 9 10 11 5 3 6 7

Inorder : 8 4 10 9 11 2 5 1 6 3 7

(i) find root from preorder

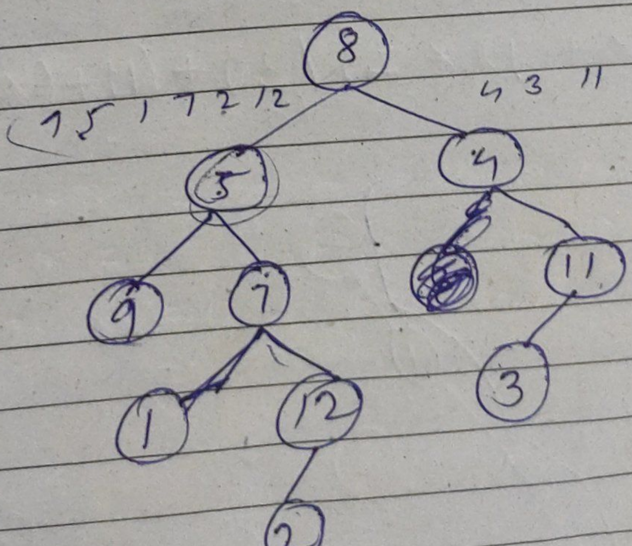


scan right left

root

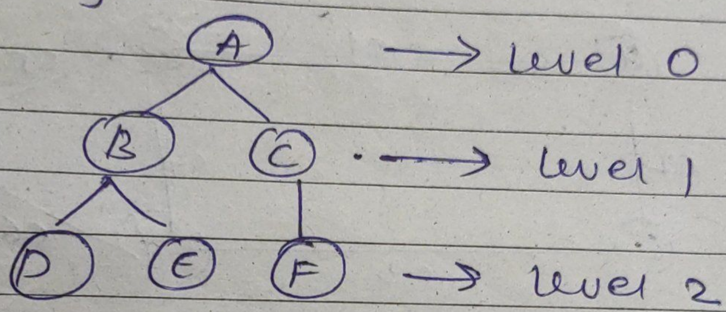
LRRoot postorder :- 9 1 2 12 7 5 3 11 4 8

LRootR Inorder :- 9 5 1 7 2 12 8 4 3 11



Rainbow
PAGE:
DATE: / /

* Level of trees :-
- Level of tree is represented by no. of stages from root node to leaf node
- On each node there are some nodes available.
- ~~Generally~~

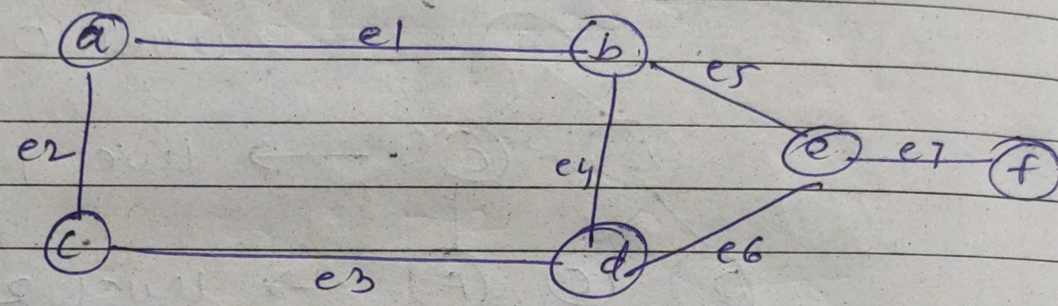


* Depth of node :-
- Depth of node is no. of edges from node to root node of the tree
- We find depth of all leaf nodes

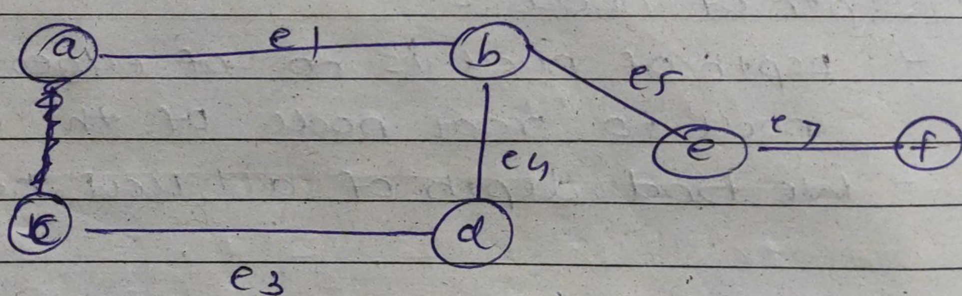
* Height of trees :-
- height of tree is the no. of edges ~~from~~ on the longest downward path betⁿ the root node and leaf node

* Forest :-
- It is a collection of disjoint trees.
- forest is a collection of acyclic graph which is not connected.
↓
No cycle

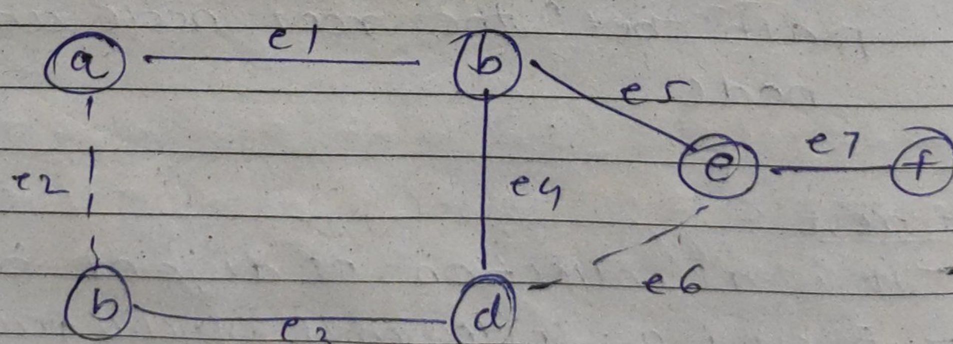
- **Fundamental circuit :-**
  - when a circuit is formed by adding chord to spanning tree i.e. (T + e), then it is called as fundamental circuit.



Graph G



Spanning tree of G



spanning tree with fundamental circuit.

* Properties of binary operation

① closure law } Algebraic structure } semi groups } monoicy } groups } Abelian groups
② Associative law
③ Indentity law
④ Inverse law
⑤ commutative law

① Closure property:-

$\boxed{\forall a, b \in A \quad a * b \in A}$ A - no empty set

$\forall$ all

eg. $\boxed{2+1=3}$

② Associative property:-

$\boxed{(a*b)*c = a*(b*c)}$

eg. $(2*1)*3 = 2*(1*3)$

$\boxed{6 = 6}$

③ Identity property:-

$\boxed{a*e = e*a = a}$

eg. $3 \times 1 = 1 \times 3 = 3$

④ Inverse property:-

$a * a^{-1} = e$

⑤ Commutative property:-

$a * b = b * a$

Abelian group

① Let G be set of all non-zero elements

→ Given

(i) $(a * b) = ab/2 \in G$ as $ab \neq 0$

$\therefore$ It satisfies closure property.

LHS=RHS (ii) Associative property.

$$(a * b) = \frac{ab}{2}$$

$$(a * b) * c = a * (b * c)$$

LHS

$(a * b) * c$

$= \frac{ab}{2} * c$

$= \frac{abc}{4}$

RHS

$a * (b * c)$

$\Rightarrow a * \frac{bc}{2}$

$\Rightarrow \frac{abc}{2 \times 2} \Rightarrow \frac{abc}{4}$

$\therefore$ $*$ is associative in G.

Find identity element (iii) Identity property

$a \times e = a$ $\qquad$ $e \times a = a$

$\frac{ae}{2} = a$ $\qquad$ $\frac{ea}{2} = a$

$e = 2$ $\qquad$ $e = 2$

$\therefore$ 2 is identity element in G

Rainbow

PAGE:

DATE: / /

Fraction (iii) Inverse property

$$\overset{a}{a} \times \overset{b}{a^{-1}} = e$$

$$a \times a^{-1} = \frac{aa^{-1}}{2} = e$$

$$2 = \frac{aa^{-1}}{2}$$

$$aa^{-1} = 4$$

$$\boxed{a^{-1} = \frac{4}{a}}$$

$\therefore$ Inverse of a is $\frac{4}{a}$, $\forall a \in G$.

(iv) Commutative property:

$$a * b = b * a$$

$$\frac{ab}{2} = \frac{ba}{2}$$

$\therefore$ LHS = RHS

$\therefore$ It satisfies commutative property.

Thus, $(G, *)$ is an abelian group.

Rainbow
PAGE:
DATE: / /

Ring

The structure $(R, +, \times)$ consisting of non-void set R and two binary composition, denoted by + and × or. Is said to be ring. if the folowing axioms satisfies:-

(i) $(R, +)$ is an albelian group with identity 0

(ii) $(R, \cdot)$ is semigroup means it
(iii)

* Types of Ring.

1) Commutative Ring
   - $(R, +, \cdot)$
   - $a \cdot b = b \cdot a$

ii) Ring with unity.
   - $(R, +, \cdot)$
   - if it has multiplicative identity
   - : $a \cdot e = a = e a$
   - $\boxed{e = 1}$

iii) Ring with zero divisor
   - ~~$a \cdot b \neq 0$~~ $a \cdot b = 0$
   - if $a \neq 0$ & $b \neq 0$
   - $5 *_3 6 = \frac{5 \times 6}{3} = \frac{30}{3} = 10$

Remider. = 0

iv) Ring without zero divisor.

$a \cdot b \neq 0$

$a \neq 0$, $b \neq 0$

* Integral Domain

- $R(+, \cdot)$
-

i) Ring → commutative Ring

$a \cdot b = b \cdot a$

ii) Ring → nonzero Divisor Ring w.r.t · operator

$a \cdot b \neq 0$

$a \neq 0$ $b \neq 0$

iii) Ring → should be Distributive w.r.t (+)

- $a * (b+c) = a*b + a*c$
- $(b+c) * a = b*a + c*a$

* Field

- $R(+, \cdot)$
-

(i) $(\cdot)$ operator should be commutative Ring

- $a \cdot b = b \cdot a$

(ii) $(\cdot)$ operator should be Distributive over addition (+)

- $a * (b+c) = a*b + a*c$
- $(b+c) * a$ $b*a + c*a$

(iii) Non zero Divisor Ring w.r.t $(\cdot)$ operator $a \cdot b \neq 0$

$a \neq 0$ $b \neq 0$

division Ring = multiplication Inverse

PAGE:
DATE: / /

(iv) Non zero element they should have multiplicative inverse.

Field = Integral Domain + divison Ring

* Define

1] Field :-

A commutative ring with unity in which every non zero element possesses their multiplicative inverse, is called as field. A field is an integral domain.

eg. - $(R, +, \cdot)$, $(Q, +, \cdot)$ are fields
- $(Z, +, \cdot)$ is integral domain but not field.

2] Ring :-

The structure $(R, +, \cdot)$ consisting of non-empty set R and two binary composition denoted by + and x or . is said to be ring if the following axioms satisfies :-

(i) $(R, +)$ is an abelian group
(ii) $(R, \cdot)$ is a semigroup
(iii) For any 3 elements $a, b, c \in R$ the left distributive law $a \cdot (b + c) = a \cdot b + a \cdot c$ and the right distributive property $(b + c) \cdot a = ab + c \cdot a$ holds.

3] Ring Homomorphism :-

- Let $(R, +, *)$ and $(S, +, *)$ be two rings.
- A Function $\phi : R \to S$ is called a ring homomorphism.

4] Integral Domain :-

- A commutative ring with zero divisors is called as integral domain

eg.

1. $(R, +, \cdot)$ $(Z, +, \cdot)$ are integral domains.
2. $(Z_4, +, \cdot)$ is ring with zero divisors, $\therefore$ It is not integrate domain.

5] Ring with unity :-

- A ring $(R, +, \cdot)$ is called as ring with unity if $\forall a \in R$, $\exists\, 1 \in R$, such that $a \cdot 1 = 1 \cdot a = a$

eg.

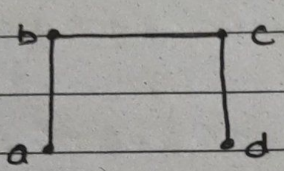
$(Z, +, \cdot)$ is commutative ring with unity.

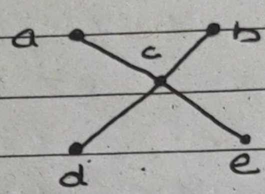
Tree

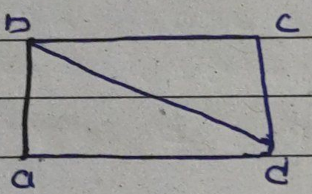
* Tree:-

- A tree is non linear structure which has many application like searching, sorting, information storage and many more.

- A tree is connected graph without a simple circuit.







-

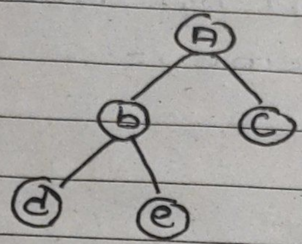
* Properties of tree:-

① A graph G is a tree if and only if there exists a unique path between every distinct pair of vertices of G

② A Graph G of n vertices is a tree if and only if G is connected and has exactly (n-1) edges.

* Basic Termonologies :-

①



① Root / [redacted] :-

- The Root node is an origin of tree.
- It does not have parent all nodes are acessed using this node
- eg - A.

② Parent :-

- Parent node is immediate predecessor of that node.
- For example B is parent node of d and e.

③ Child node :-

- Child node is immediate succe ssor of any node
- For example d and e is child of b.

④ Siblings :-

- Sibling are nodes with same parent
- For example d and e is sibling

[redacted]

⑤ **Leaf Node:-**

- The leaf node is node with no children element.
- For eg- d, e, c.

⑥ **Internal Node:-**

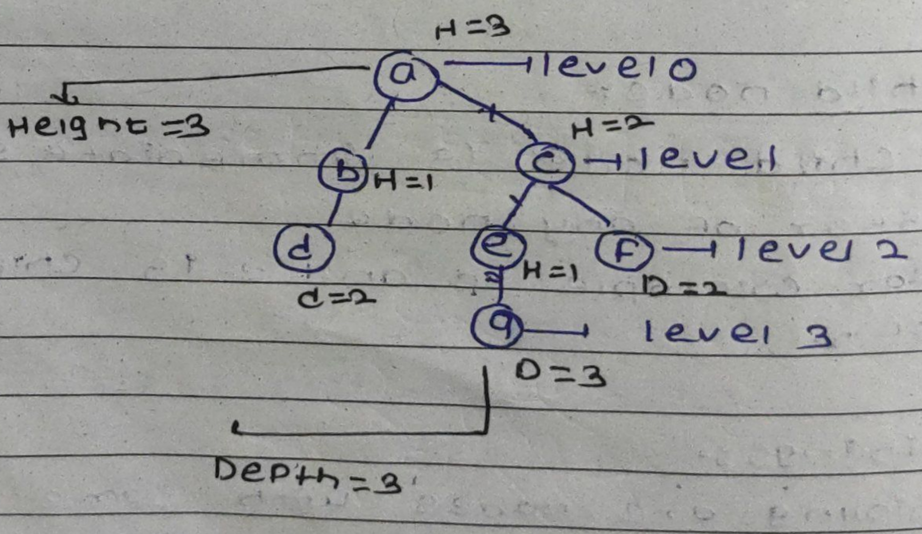
- The Nodes present between Root node and leaf node is called internal Node.

⑦ **Degree of Node:**

- The no of child node connected to it is called degree of Node.
- For example A has 2 degree, b has 2 degree, d, e, f has 0 degree.

⑧ **level of Node:-**

- The level of node is represented by number of stage from root node to leaf node.



Rainbow
PAGE:
DATE: / /

(9) Height :-

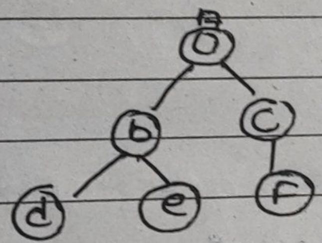
- The height of tree is number of edges on longest downword path between root node and leaf node

(10) Depth :-

- The No of edges from node to root node.

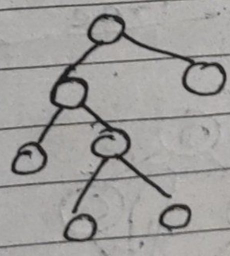
* Binary Tree :-

- A binary tree is tree in which every node or vertex has atmost 2 child.



* Full / Strict Binary tree :-

- Internal node have exactly 2 child.



* Complete Binary :-

- Each node have two child but leaf node with same level.