Logic And Proofs unit - 1

Proposition :-

A proposition is a declarative sentence that is either true or false, but not both.

Eg: Find whether the following declarative sentence or proposition.

1. Washington, D.C is the capital of the U.S.A
2. Hyderabad is the capital of kerela.
3. $1+1=2$
4. $2+2=3$

Sol: From the above 4 propositions, 1 and 3 are true whereas 2 and 4 are false.

Negation of P:

Let p be a proposition. The statement "It is not the case that p" is another proposition, called the negative of p. The negative of p is denoted by $\neg p$. The proposition $\neg p$ is read not p.

Eg: Find the negation of the proposition "Today is friday" and express this in simple english.

Sol: The Negation is "It is not the case that today is friday."

This negation can be most simply expressed as "Today is not Friday" (or) it is not friday today.

The truth table from the negative of proposition.

p	$\neg p$	$\neg(\neg p)$
T	F	T
F	T	F

Truth table :

A truth table displays the relations between the truth values of propositions. They can determine the truth values of propositions constructed from simple propositions.

Types of conjugate :

There are four types of conjugate.

1) conjunction
2) Disjunction
3) Implication
4) Bi- implication

1) Conjunction :

If p and q are any two proposition then the conjunction of p and q is given by the symbol "$p \wedge q$". In case of these conjunction if both p and q are true then the conjugate conjunction is true and false in remaining all other cases.

The truth table is given by

p	q	$p \wedge q$
T	T	T
T	F	F
F	T	F
F	F	F

2) Disjunction :

If p and q are any two statements then the disjunction of p or q is given by the symbol "$p \vee q$". If p and q both are false then the $p \vee q$ is false. In remaining other cases the conjugate disjunction is true.

The truth table is given by

p	q	$p \vee q$
T	T	T
T	F	T
F	T	T
F	F	F

3) Implication :

If p and q are any two statements then the implication of p and q is given by the symbols "$p \rightarrow q$" (or) "$p \Rightarrow q$". If p is true, q is false then $p \Rightarrow q$ is false and remaining all cases it is true

The truth table is given by

p	q	$p \Rightarrow q$
T	T	T
T	F	F
F	T	T
F	F	T

The implication can be expressed as

"If p then q". "p implies q"

"If p,q" "p only if q".

"p is sufficient"   "a sufficient condition
"q if p"   for q is p"
"q when p"   "q whenever p"
"a necessary condition   "q is necessary for p"
for p is q"   "q follows from p"

4) Bi-implication :

If p, and q are any two statements then bi-implication p and q is given by $p \Leftrightarrow q$. In this case if p and q are both true. $p \Leftrightarrow q$ is true. If p and q are both false then $p \Leftrightarrow q$ is true.

p	q	$p \Leftrightarrow q$
T	T	T
T	F	F
F	T	F
F	F	T

The truth table is given above
$\therefore$ $p \Leftrightarrow q$ means "p if and only if q"

Logical Equivalence :

Two statements are said to be logically equivalent if they have identical truth tables:

Eg :

1) Show that the statements $p \to q$, and $\neg p \vee q$ are logically equivalent.

Sol: We have to proof $p \to q$ and $\neg p \vee q$ are logically equivalent we have to construct the truth table.

p	q	$p \Rightarrow q$	$\neg p$	$\neg p \vee q$
T	T	T	F	T
T	F	F	F	F
F	T	T	T	T
F	F	T	T	T

Since the truth value of $p \Rightarrow q$ and $\neg p \vee q$ are identical the two statements are logically equivalent

2) Construct truth tables from following

(i) $(A \wedge B) \wedge (\neg B \vee C)$ (ii) $A \Leftrightarrow (B \Leftrightarrow C)$

Sol (i)

A	B	C	$A \wedge B$	$\neg B$	$\neg B \vee C$	$(A \wedge B) \wedge (\neg B \vee C)$
T	T	T	T	F	T	T
T	T	F	T	F	F	F
T	F	T	F	T	T	F
T	F	F	F	T	T	F
F	T	T	F	F	T	F
F	T	F	F	F	T	F
F	F	T	F	T	T	F
F	F	F	F	T	T	F

(ii)

A	B	C	$B \Leftrightarrow C$	$A \Leftrightarrow (B \Leftrightarrow C)$
T	T	T	T	T
T	T	F	F	F
T	F	T	F	F
T	F	F	T	T
F	T	T	T	F
F	T	F	F	T
F	F	T	F	T
F	F	F	T	F

Tautology :-

A compound proposition that is always true, no matter what the truth values of the propositions that occur in it is called tautology.

Eg:

p	$\neg p$	$p \vee \neg p$
T	F	T
F	T	T

Contradiction :-

A compound proposition that is always false is called a contradiction.

p	$\neg p$	$p \wedge \neg p$
T	F	F
F	T	F

Contingency :

A proposition that is neither a tautology nor a contradiction is called contingency.

1) Show that $(p \wedge q) \rightarrow (p \vee q)$ is tautology.

p	q	$p \wedge q$	$p \vee q$	$(p \wedge q) \rightarrow (p \vee q)$
T	T	T	T	T
T	F	F	T	T
F	T	F	T	T
F	F	F	F	T

Bitwise Operators :-

A bit has two possible values

namely 0 and 1. A variable is called boolean variable if its value is either true or false. Consequently, a boolean variable can be represented using a bit. computer bit operations corresponds to the logical connectives. By replacing a true by 1 and false by 0. In the truth table for the operations OR, AND, XOR.

p	q	$p \vee q$	$p \wedge q$	$p \oplus q$
0	0	0	0	0
0	1	1	0	1
1	0	1	0	1
1	1	1	1	0

Logical Equivalences:

Equivalence	Name
$p \wedge T \equiv p$ $p \vee F \equiv p$	Identity laws
$p \vee T \equiv T$ $p \wedge F \equiv F$	Domination laws
$p \vee p \equiv p$ $p \wedge p \equiv p$	Idempotent laws
$\neg(\neg p) \equiv p$	Double Negation laws
$p \vee q \equiv q \vee p$ $p \wedge q \equiv q \wedge p$	Commutative laws
$(p \vee q) \vee r \equiv p \vee (q \vee r)$ $(p \wedge q) \wedge r \equiv p \wedge (q \wedge r)$	Associative laws
$p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r)$ $p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r)$	Distributive laws
$\neg(p \wedge q) \equiv \neg p \vee \neg q$ $\neg(p \vee q) \equiv \neg p \wedge \neg q$	Demorgan's theorem

$p \lor (p \land q) \equiv p$ Absorption laws

$p \land (p \lor q) \equiv p$

$p \lor \neg p \equiv T$ Negation laws

$p \land \neg p \equiv F$

Eg:

1) verify the following commutative laws by using truth tables.

(i) $p \lor q \equiv q \lor p$ (ii) $p \land q \equiv q \land p$

Sol:

p	q	$p \lor q$	$q \lor p$
T	T	T	T
T	F	T	T
F	T	T	T
F	F	F	F

p	q	$p \land q$	$q \land p$
T	T	T	T
T	F	F	F
F	T	F	F
F	F	F	F

$\therefore p \lor q \equiv q \lor p$ $\therefore p \land q \equiv q \land p$

$\therefore$ commutative laws are verified

2) Verify Demorgan's law by using truth tables

(i) $\neg(p \land q) \equiv \neg p \lor \neg q$ (ii) $\neg(p \lor q) \equiv \neg p \land \neg q$.

Sol: (i)

p	q	$\neg p$	$\neg q$	$\neg(p \land q)$	$\neg p \lor \neg q$
T	T	F	F	F	F
T	F	F	T	T	T
F	T	T	F	T	T
F	F	T	T	T	T

$\therefore \neg(p \land q) \equiv \neg p \lor \neg q$

(ii)

p	q	$\neg p$	$\neg q$	$\neg(p \lor q)$	$\neg p \lor \neg q$
T	T	F	F	F	F
T	F	F	T	F	F
F	T	T	F	F	F
F	F	T	T	T	T

$\therefore \neg(p \vee q) \equiv (\neg p \vee \neg q)$

$\therefore$ Demorgan's laws are verified.

Show that $\neg(p \vee (\neg p \wedge q))$ and $\neg p \wedge \neg q$ are logically equivalent.

$\neg(p \vee (\neg p \wedge q))$

$\Rightarrow \neg p \wedge (\neg \neg p \wedge q)$ (Demorgans law)

$\Rightarrow \neg p \wedge (p \vee \neg q)$ (" ")

$\Rightarrow (\neg p \wedge p) \vee (\neg p \wedge \neg q)$ (Distributive law)

$\Rightarrow F \vee (\neg p \wedge \neg q)$ (Identity law)

$\Rightarrow \neg p \wedge \neg q$

Show that $(p \wedge q) \rightarrow (p \vee q)$ is a tautology.

$(p \wedge q) \rightarrow (p \vee q)$

$\Rightarrow \neg(p \wedge q) \vee (p \vee q)$ (By demorgan's law)

$\Rightarrow (\neg p \vee \neg q) \vee (p \vee q)$

$\Rightarrow (\neg p \vee p) \vee (\neg q \vee q)$ (By Commutative and Associative law)

$\Rightarrow T \vee T$

$\Rightarrow T$

$\therefore (p \wedge q) \rightarrow (p \vee q)$ is a tautology

logical Equivalences involving implications

$p \rightarrow q \equiv \neg p \vee q$

$p \rightarrow q \equiv \neg q \rightarrow \neg p$

$p \vee q \equiv \neg p \rightarrow q$

$p \wedge q \equiv \neg(p \rightarrow \neg q)$

$\neg(p \rightarrow q) \equiv p \wedge \neg q$

$(p \rightarrow q) \wedge (p \rightarrow r) \equiv p \rightarrow (q \wedge r)$

$(p \rightarrow r) \wedge (q \rightarrow r) \equiv (p \vee q) \rightarrow r$

$(p \rightarrow q) \vee (p \rightarrow r) \equiv p \rightarrow (q \vee r)$

$(p \rightarrow r) \vee (q \rightarrow r) \equiv (p \wedge q) \rightarrow r$

Determine wheather $(\neg P \wedge (P \to q)) \to \neg q$ is a tautology

$$(\neg P \wedge (P \to q)) \to \neg q$$

$$\Rightarrow \neg(\neg P \wedge (P \to q)) \vee \neg(\neg q)$$

$$\Rightarrow \neg(\neg P \wedge (\neg P \vee q)) \vee \neg(\neg q)$$

$$\Rightarrow (\neg(\neg P) \vee \neg(\neg P \vee q)) \vee q$$

$$\Rightarrow P \vee (\neg(\neg P) \wedge \neg q) \vee q$$

$$\Rightarrow (P \vee P) \wedge (\neg q \vee q)$$

$$\Rightarrow P \wedge T$$

$$\Rightarrow P$$

Conjuctive Normal Form :

A logical formula consisting of a conjunction of disjunction of terms where disjunction contains a conjunction such a formula might also be describe as a product of sums

Eg : (A and B) or C is (A or C) and (B or C)

i.e., $(A \wedge B) \vee (B \wedge C)$ $(A \vee C) \wedge (B \vee C)$

Disjunctive Normal Form :

A logical formula consisting of a disjunction of conjunctions where no conjunction contains a disjunction

(A or B) and C is (A and C) or (B and C)

i.e., $(A \wedge C) \vee (B \wedge C)$

Eg :

1) Obtained a disjunctive normal form for the following conjugative statement $(P \wedge \neg(Q \wedge R)) \vee (P \to q)$

Sol: $(P \wedge \neg(Q \wedge R)) \vee (P \to Q)$

According to the definition of disjunctive normal form, the formula which is equal to the given formula and which consists a sum of elementary products.

$R \Leftrightarrow S \equiv (R \wedge S) \vee (\neg R \wedge \neg S)$

We have to convert the given probability the above pattern

$\Rightarrow (P \wedge \neg(Q \wedge R)) \vee (P \rightarrow Q)$

$\Rightarrow (P \wedge (\neg Q \vee \neg R)) \vee (\neg P \vee Q)$

$\Rightarrow (P \wedge \neg Q) \vee (P \wedge \neg R) \vee (\neg P \vee Q)$

$\Rightarrow (P \wedge \neg Q) \vee (P \wedge \neg R) \vee \neg(\neg P \wedge \neg Q)$

2) Obtained a disjunctive Normal form for the following compound statements and also check whether it may be a tautology: $P \vee [\neg P \Rightarrow (Q \vee (Q \Rightarrow \neg P))]$

Sol

$P \vee [\neg P \Rightarrow (Q \vee (Q \Rightarrow \neg P))]$

$\Rightarrow P \vee [P \vee (Q \vee (Q \Rightarrow \neg P))]$

$\Rightarrow P \vee [P \vee (Q \vee (\neg Q \vee \neg P)]$

$\Rightarrow P \vee P \vee (Q \vee (\neg Q \vee \neg P)]$

$\Rightarrow P \vee (Q \vee \neg Q) \vee \neg P$

$\Rightarrow (P \vee \neg P) \vee T$

$\Rightarrow T \vee T$

$\Rightarrow T$

$\therefore P \vee [\neg P \Rightarrow (Q \vee (Q \Rightarrow \neg P))]$ is a tautology.

Logical Equivalences involving Bi-implications (or) Bi-conditionals :-

$P \Leftrightarrow q \equiv (P \rightarrow q) \wedge (q \rightarrow P)$

$P \Leftrightarrow q \equiv (\neg P \Leftrightarrow \neg q$

$P \Leftrightarrow q \equiv (P \wedge q) \vee (\neg P \wedge \neg q)$

$\neg(P \Leftrightarrow q) \equiv P \Leftrightarrow \neg q$

Eg :-

1) Find the conjunctive normal form for the following compound statements.

$P \wedge (P \to Q) \Leftrightarrow P \wedge (\neg P \vee Q)$

Sol: According to the definition of Conjunctive normal form, a formula which is equal to the given formula and consists of product of sums. From Bi-implication logical equivalence

$P \Leftrightarrow Q \equiv (P \to Q) \wedge (Q \to P)$ then

$[(P \wedge (P \to Q)) \to (P \wedge (\neg P \vee Q))] \wedge [(P \wedge (\neg P \vee Q)) \to (P \wedge (P \to Q))]$

$\Rightarrow [P \wedge (\neg P \vee Q) \to (P \wedge (\neg P \vee Q))] \wedge [(P \wedge (\neg P \vee Q)) \to P \wedge (\neg P \vee Q)]$

$\Rightarrow [P \wedge (\neg P \vee Q) \to P \wedge (\neg P \vee Q)]$

$\Rightarrow [\neg (P \wedge (\neg P \vee Q)) \vee (P \wedge (\neg P \vee Q))]$

$\Rightarrow \neg [(P \wedge \neg P) \vee (P \wedge Q)] \vee [(P \wedge \neg P) \vee (P \wedge Q)]$

$\Rightarrow \neg (P \wedge Q) \vee (P \wedge Q)$

$\Rightarrow (\neg P \vee \neg Q) \vee \neg (\neg P \wedge \neg Q)$

$\Rightarrow \neg (\neg P \vee \neg Q) \wedge \neg (\neg (\neg P \vee \neg Q))$

$\Rightarrow \neg (\neg P \vee \neg Q) \wedge (\neg P \vee \neg Q)$

Principle Conjunctive Normal Forms :-

A Formula which is equivalent to given formula and consisting of conjunction to the maximum number of terms, is only know as principle Conjunctive normal Form (PCNF)

Principle Disjunctive Normal Form :-

A Formula which is equivalent to given formula and Consisting of disjunctions to the minimum number of terms is only known as Principle disjunctive normal Form (PDNF)

Eg:-

1) Obtained the PDNF to the compound statement $(\neg P \vee Q)$

Sol: $\neg P \vee Q \equiv [\neg P \wedge (Q \vee \neg Q)] \vee [Q \wedge (P \vee \neg P)]$

$\equiv (\neg P \wedge Q) \vee (\neg P \wedge \neg Q) \vee (Q \wedge P) \vee (Q \wedge \neg P)$

$\equiv (\neg P \wedge Q) \vee (\neg P \wedge \neg Q) \vee (Q \wedge P)$

2) find PCNF of the following statements $[P \wedge (P \to Q)] \Leftrightarrow P \wedge (\neg P \vee Q)$.

Sol: $[P \wedge (P \to Q)] \Leftrightarrow P \wedge (\neg P \vee Q)$

With respect to Bi-implication logical Equivalence

$P \Leftrightarrow Q \equiv (P \to Q) \wedge (Q \to P)$

$[P \wedge (P \to Q)] \Leftrightarrow P \wedge (\neg P \vee Q)$

$\equiv [P \wedge (P \to Q) \to P \wedge (\neg P \vee Q)] \wedge [P \wedge (\neg P \vee Q) \to P \wedge (P \to Q)]$

$\equiv (\neg (P \wedge (\neg P \vee Q)) \vee (P \wedge (\neg P \vee Q))] \wedge [\neg (P \wedge (\neg P \vee Q)) \vee P \wedge (\neg P \vee Q)]$

$\equiv \neg ((P \wedge \neg P) \wedge (P \wedge Q)] \vee [(P \wedge \neg P) \vee (P \wedge Q)] \wedge \neg [(P \wedge \neg P) \vee (P \wedge Q)] \vee [(P \wedge \neg P) \vee (P \wedge Q)]$

$\equiv \neg (P \wedge Q) \vee (P \wedge Q) \wedge \neg (P \wedge Q) \vee (P \wedge Q)$

$\equiv \neg (P \wedge Q) \vee (P \wedge Q)$.

$\equiv (\neg P \vee \neg Q) \vee (P \wedge Q)$

Predicate :-

A predicate is a property that the subject of the statement can have, we denote the statement "$x > 3$" by $P(x)$, where P is predicate is greater than 3 and x is the variable. This statement $P(x)$ also said to be the value of the propositional function P at x

Eg:

1) Let $P(x)$ denote the statement $x > 3$, what are the truth values of $P(4)$ and $P(2)$

Sol: We obtained the statement $P(4)$ by setting $x = 4$ in the statement "$x > 3$". Hence, $P(4)$ is true since, $4 > 3$ where as $P(2)$ is False since, $2 \not> 3$.

2) Find canonical sum of products Normal form to the below compound statements.

(i) $P \rightarrow (P \rightarrow Q) \wedge \neg(\neg Q \wedge \neg P)$

(ii) $\neg(P \vee Q) \equiv \neg P \wedge \neg Q$.

Sol (i)

$$P \rightarrow (P \rightarrow Q) \wedge \neg(\neg Q \wedge \neg P)$$

$$\equiv \neg P \vee (P \rightarrow Q) \wedge \neg(\neg Q \wedge \neg P)$$

$$\equiv \neg P \vee [(\neg P \vee Q) \wedge (Q \vee P)]$$

$$\equiv \neg P \vee [\neg P \vee Q) \wedge (Q \vee P)$$

$$\equiv \neg P \vee [((\neg P \vee Q) \wedge Q) \vee ((\neg P \vee Q) \wedge P)]$$

$$\equiv \neg P \vee [(\neg P \wedge Q) \vee (Q \wedge Q) \vee (\neg P \wedge P) \vee (Q \wedge P)]$$

$$\equiv \neg P \vee [(\neg P \wedge Q) \vee Q \vee F \vee (Q \wedge P)]$$

$$\equiv \neg P \vee [(\neg P \wedge Q) \vee Q \vee (Q \wedge P)]$$

$$\equiv \neg P \vee [(\neg P \wedge Q) \vee Q \vee (Q \wedge P)]$$

$$\equiv \neg P \vee [Q \vee (Q \wedge \neg P) \vee (Q \wedge P)]$$

$$\equiv \neg P \vee [Q \vee (Q \wedge \neg P) \vee (Q \wedge P)]$$

$$\equiv (\neg P \vee Q) \vee (\neg P \vee (Q \wedge \neg P)) \vee (\neg P \vee (Q \wedge P)).$$

$$\equiv (\neg P \vee Q) \vee [(\neg P \vee Q) \wedge (\neg P \vee \neg P)] \vee [(\neg P \vee Q) \wedge (\neg P \vee P)]$$

$\equiv (P\wedge\neg Q)\vee[(\neg P\vee Q)\wedge\neg P\vee(\neg P\vee Q))]$

$\equiv (P\wedge\neg Q)\vee\{(\neg P\vee Q)\wedge\neg P\}$

$\equiv (\neg P\vee Q)\vee[\neg P\vee(Q\wedge\neg P)]\vee[\neg P\vee(Q\wedge P)]$

$\equiv (\neg P\vee Q)\vee[(\neg P\vee Q)\wedge(\neg P\vee\neg P)]\vee[(\neg P\vee Q)\wedge(\neg P\vee P)]$

$\equiv (\neg P\vee Q)\vee[(\neg P\vee Q)\wedge(\neg P\vee\neg P)]\vee[(\neg P\vee Q)\wedge T]$

$\equiv (\neg P\vee Q)\wedge(\neg P\vee Q)\wedge\neg P$ ....

$\equiv (\neg P\vee Q)\wedge\neg P$

$\equiv \neg[(\neg P\vee Q)\wedge\neg P]$

$\equiv (P\wedge\neg Q)\vee P$

(iii) $\neg(P\vee Q)\equiv\neg P\wedge\neg Q$

$\equiv \neg[(P\wedge(Q\vee\neg Q))\vee(Q\wedge(P\vee\neg P))]$

$\equiv \neg[((P\wedge Q)\vee(P\wedge\neg Q))]\vee[(Q\wedge P)\vee(Q\wedge\neg P))]$

$\equiv \neg[(P\wedge Q)\vee(P\wedge\neg Q)\vee(Q\wedge P)\vee(Q\wedge\neg P)]$

$\equiv \neg[(P\wedge Q)\vee(P\wedge\neg Q)\vee(Q\wedge\neg P)]$

$\equiv \neg[(P\wedge Q)\vee(P\vee Q)\wedge(P\wedge\neg P)\wedge(\neg Q\vee Q)\wedge(\neg Q\wedge\neg P)]$

$\equiv \neg[(P\wedge Q)\vee(P\vee Q)\wedge F\wedge T\wedge(\neg Q\wedge\neg P)]$

$\equiv \neg[(P\wedge Q)\vee(P\vee Q)\wedge F\wedge(\neg Q\wedge\neg P)]$

$\equiv \neg[(P\wedge Q)\vee(P\vee Q)\wedge F]$

$\equiv \neg[(P\wedge Q)\vee F]$

$\equiv \neg[P\wedge Q]$

$\equiv \neg P\vee\neg Q$

3) Let $Q(x,y)$ denote the statement $x=y+3$. What are the truth values of proposition $Q(1,2)$ and $Q(3,0)$

Here, $Q(1,2)$ is false, since $1=2+3$ is false

$Q(3,0)$ is True, since $3 = 0+3$ is true.

Quantifier :-

Quantifiers are used to create the proposition from a propositional function called as quantification. There are two types of Quantifiers called as universal quantifiers and existential quantifier

Universal Quantifier :-

Many Mathematical Statements asserted that the property true for all values of variables in a particular domain called the universe of discourse or domain such a Statement is expressed using a universal quantification

The universal quantification of $P(x)$ is true for all values of x in the universe of discourse;

The notation is $\forall x\, P(x)$. Here $\forall$ is called universal quantifier.

Eg :-

1) Let $P(x)$ is the Statement "$x+1 > x$" what is the truth value of the quantification $\forall x\, P(x)$ where the universe of discourse consist of all real numbers.

Sol:- Since $P(x)$ is true for all real numbers x $\forall x P(x)$ is true.

2) Let $Q(x)$ be the statement "$x<2$" what is the truth value of the quantification $\forall x\, Q(x)$ where the universe of discourse consists of all real numbers.

Sol: $Q(x)$ is not true for every real number x. Since, for instance $Q(3)$ is false. Thus, $\forall x Q(x)$ is false.

3) What does the statement $\forall x T(x)$ means that for every person x. that person has two parents and the domain consists of all people

The student can be expressed as every person has two parents. It is true.

Existential quantifier :-

Any Mathematical statements, assets there is a element with a second property such statements are expressed using existential quantifiers. They existential quantification of $P(x)$ is the proposition "$\exists$ an element x is the universe of discourse such that $P(x)$ is true". For this we use the notation $\exists x P(x)$. Here, the symbol $\exists$ is called the existential quantifier.

eg:-

1) Let $P(x)$ denotes the statement "$x > 3$" what is the truth value of $\exists x P(x)$ where the universe of discourse is all real numbers.

Sol:- Since, $x > 3$ is true for instance when $x = 4$ then $\exists x P(x)$ is true.

2) What is the truth value of $\exists x P(x)$ where $P(x)$ is "$x^2 > 10$" the universe of discourse consists of the positive integers not exceeding 4.

Sd:- The universe of discourse is $\{1,2,3,4\}$ then the proposition $\exists x P(x)$ is the same as $P(x_4)$ the disjunction $P(1) \vee P(2) \vee P(3) \vee P(4)$. Since, $P(4)$ i.e $4^2 > 10$ is true. It follows that $\exists x P(x)$ is true.

Binding variables :-

A quantifier is used on the variable x. This say that this occur and properties variable is bound. An occurance of a variable that is not bound by a quantifier is said to be free. The part of the logical expressions to which a quantifier is obtained is called the scope of the quantifier.

Therefore, a variable is free if it is outside the scope of all quantifiers in the formula that specifier this variable.

Eg :-

$\exists x\, Q(x,y)$, Here x is bounded by the existential quantifier $\exists$ but the variable y is free. Because it is not bounded by any quantifier.

Eg :-

$\exists x\, (P(x) \wedge Q(x)) \forall x R(x)$, find the bounded variables and scope of the variables.

Sd:- Here, all the variables are bounded. The scope of the first quantifier $\exists x$ is that expression $P(x) \wedge Q(x)$. Because it is obtained own to the expression and not to be the established statement. Similarly, the scope of the second quantifier $\forall x$ is the expression $R(x)$

Negations :-

(1) $\neg \forall x P(x) \equiv \exists x \neg P(x)$

(2) $\neg \exists x P(x) \equiv \forall x \neg P(x)$

Eg :-

1) what are the negations of the statements "There is an honest politician" and "All Americans eat cheese burgers".

Sol Let $h(x)$ denotes "x is honest" then the statement "There is an honest politician" is represented by $\exists x h(x)$, where the universe of discourse consists of all politicians then the Negation of this statement is $\neg(\exists x h(x)) \equiv \forall x \neg h(x)$. i.e., All politicians are dishonest.

Let $C(x)$ denotes "x eats cheese burgers". Then the statement "All Americans eat cheese burgers". Here C is represented by $\forall x(x)$; where the universe of discourse consists of all americans. Then the Negation of this statement is $\neg(\forall x C(x)) \equiv \exists x \neg C(x)$ i.e., There is an American who does not eat cheese burgers.

Translating from English into logical Expression :

Converting a propositional statement in english type language into a predicate with quantifiers.

Eg:-

Express the statement "every student in this class has studied calculus" using predicates and quantifiers.

Sol: For this we need to introduce a variable x. So, that our statement becomes for every student x in this class, x has studied calculas. For this we introduce the predicate $C(x)$ which is the statement "x has studied calculus". If we consider the universe of discourse for x, consists of the students in the class, we can translate the statement has $\forall x\, C(x)$

If we consider the universe of discourse for x to consists of all people we need to express our statement as:

"For every person x, if person is a student in the class then x has studied calculus". If $S(x)$ represent the statement that "Person x is in this class". We can express the statement has $\forall x\, S(x) \rightarrow C(x)$

Nested Quantifiers:-

The Quantifiers that occur with in the scope of other quantifiers such as in the statement $\forall x \exists y (x+y=0)$.

Eg:-

1.) Translate into english statement $\forall x \forall y ((x>0) \wedge (y<0) \rightarrow (xy<0))$, where the universe of discourse for both variables consists of all real number

Sol: This statement says that for every real Number x and for every real number y, if $x>0$ and $y<0$ then $xy<0$ i.e, this statement says that for all real number x and y. If x is positive and y is negative then xy is negative. This statement can be expressed as "The product of a positive real number and negative real number is negative real number.

(2) Translate the statement $\forall x\,[C(x) \vee \exists y\,(C(y) \wedge F(x,y))]$ into english, where $C(x)$ is "x has a computer", $F(x,y)$ is "x and y are friends", and the universe of discourse for both x and y consists of all students in your school.

Sol: The statement says that for every student x in your school x has a computer or there is a student y such that y has a computer and x and y are friends. In other words every student in your school has a computer or has a friend who has a computer.

Order of Quantifiers :-

Eg:

let $Q(x,y)$ denotes "$x+y=0$" what are the truth values of the quantifications $\exists y \forall x\, Q(x,y)$ and $\forall x \exists y\, Q(x,y)$.

Sol: For the quantifications $\exists y \forall x\, Q(x,y)$.

It denotes the proposition "There is a real number y such that for every real number x, $Q(x,y)$". Since, there is no real number y such that $x+y=0$. for all real numbers x, the statement $\exists y \forall x\, Q(x,y)$ is false.

The quantification $\forall x \exists y\, Q(x,y)$ denotes the proposition "for every real numbers x there is a real number y" such that $Q(x,y)$

Given a real number x, there is a real number y such that $x+y=0$ namely $y=-x$. Hence, the statement $\forall x \exists y\, Q(x,y)$ is true.

Sets and Set Operations

Set :-

Set is a collection of elements. Generally they are denoted by capital letters.

Members of a set :-

The members or letters that are involved in a given set are called members of the given set

Infinite Set :-

A set consists of infinite number of elements is called infinite set.

Void Set :-

A Set with no elements is called void set which is denoted by $\{\}$ or ϕ.

Universal Set :-

The set of all sets which comes under discussion is known as universal set. which is denoted by 'U'

Superset and Subset :-

Set A is said to be subset of B $\Leftrightarrow$ Every element of A is also an element of B which is denoted as $A \subseteq B$, Here B is the super set.

Proper subset :-

A is said to be set proper subset of B $\Leftrightarrow$ A is contained in B. But $A \neq B$

Power Set :-

Given a set S, the proper set of S is

the set of all subsets of the sets. The power set of S is denoted by P(S).

Eg:

What is power set of {0,1,2}

Sol: $P(S) = P(\{0,1,2\})$

$= \{\phi, \{0,1,2\}, \{0,1\}, \{0,2\}, \{1,2\}, \{0\}, \{1\}, \{2\}\}$

Complement Set :-

Let 'U' be the universal set. The complement of A is denoted by $\bar{A}$ or A' & the complement of A with respect to U. In otherwords the complement of the set A is $U-A$.

Operations on Sets :-

Set union :

If A,B are two sets then the union consists of all elements of A and B. It is denoted by $A \cup B$.

$$x \in A \cup B \Rightarrow x \in A \text{ or } x \in B$$

Set Intersection :

If A,B are two sets then the intersection consists of all common elements are A and B. It is represented as $A \cap B$

$$x \in A \cap B \Rightarrow x \in A \text{ and } x \in B$$

Set Difference :

Let A and B be sets, the difference of A and B is denoted by $A-B$, is the set containing those elements that all are in A but not in B. The difference of A and B is also called the complement of B with respect to A i.e., $A - B = \{x / x \in A \cap x \notin B\}$

Eg:-

1) True the following by set operations commutative and associative.

Sol:

(1) Commutative property :-

(i) $A \cup B = B \cup A$

$x \in A \cup B$

$\Rightarrow x \in A$ or $x \in B$

$\Rightarrow x \in B$ or $x \in A$

$\Rightarrow x \in B \cup A$

$\therefore A \cup B = B \cup A$

(ii) $A \cap B = B \cap A$

$x \in A \cap B$

$\Rightarrow x \in A$ and $x \in B$

$\Rightarrow x \in B$ and $x \in A$

$\Rightarrow x \in B \cap A$

$\therefore A \cap B = B \cap A$

(2) Associative property :-

(i) $(A \cup B) \cup C = A \cup (B \cup C)$

$\Rightarrow x \in A \cup B$ (or) $x \in C$

$\Rightarrow x \in A$ or $x \in B$ or $x \in C$

$\Rightarrow x \in A$ or $x \in B \cup C$

$\Rightarrow x \in A \cup (B \cup C)$

$\therefore (A \cup B) \cup C = A \cup (B \cup C)$

(ii) $(A \cap B) \cap C = A \cap (B \cap C)$

$\Rightarrow x \in (A \cap B)$ and $x \in C$

$\Rightarrow x \in A$ and $x \in B$ and $x \in C$

$\Rightarrow x \in A$ and $x \in B \cap C$

$\Rightarrow x \in A \cap (B \cap C)$

$\therefore (A \cap B) \cap C = A \cap (B \cap C)$

(3) Distributive property :-

(i) $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$

$\Rightarrow x \in A \cup (B \cap C)$

$\Rightarrow x \in A$ or $x \in (B \cap C)$

$\Rightarrow x \in A$ or ($x \in B$ and $x \in C$)

$\Rightarrow$ ($x \in A$ or $x \in B$) and ($x \in A$ or $x \in C$)

(ii) $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$

$\Rightarrow x \in A \cap (B \cup C)$

$\Rightarrow x \in A$ and $x \in (B \cup C)$

$\Rightarrow x \in A$ and ($x \in B$ or $x \in C$)

$\Rightarrow$ ($x \in A$ and $x \in B$) or ($x \in A$ and $x \in C$)

$\Rightarrow x \in A \cup B$ and $x \in A \cup C$

$\Rightarrow x \in (A \cup B) \cap (A \cup C)$

$\therefore A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$

$\Rightarrow x \in (A \cap B)$ or $x \in (A \cap C)$

$\Rightarrow x \in (A \cap B) \cup (A \cap C)$

$\therefore A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$

1\. Prove that $A - (B \cap C) = (A - B) \cup (A - C)$

$x \in A - (B \cap C)$

$\Leftrightarrow x \in A$ and $x \notin (B \cap C)$

$\Leftrightarrow x \in A$ and $(x \notin B$ or $x \notin C)$

$\Leftrightarrow (x \in A$ and $x \notin B)$ or $(x \in A$ and $x \notin C)$

$\Leftrightarrow x \in (A - B) \cup x \in (A - C)$

$\Leftrightarrow x \in (A - B) \cup (A - C)$

$\therefore A - (B \cup C) = (A - B) \cup (A - C)$

2\. $A - (B \cup C) = (A - B) \cap (A - C)$

$x \in A - (B \cup C)$

$\Rightarrow x \in A$ and $x \notin (B \cup C)$

$\Rightarrow x \in A$ and $(x \notin B$ and $x \notin C)$

$\Rightarrow (x \in A$ and $x \notin B)$ and $(x \in A$ and $x \notin C)$

$\Rightarrow x \in (A - B)$ and $x \in (A - C)$

$\Rightarrow x \in (A - B) \cap (A - C)$

$\therefore A - (B \cup C) = (A - B) \cap (A - C)$

Indexed Set, Index Set, Index :–

Let $J = \{S_1, S_2, \dots S_n\}$, $A = \{AS_1, AS_2, \dots AS_n\}$

If $AS_i = AS_j \Leftrightarrow S_i = S_j$ where $S_i, S_j \in J$. Therefore the above set A is called indexed set, J is called index set and AS_i is called the Index.

(1) prove the following.

1) $(A \cup B)' \subset A' \cap B'$

let $x \in (A \cup B)'$

$\Rightarrow x \in (U - (A \cup B))$

$\Rightarrow x \in U$ and $x \in \neg(A \cup B)$

$\Rightarrow x \in U$ and $(x \in A'$ and $x \in B')$

$\Rightarrow (x \in U$ and $x \in A')$ and $(x \in U$ and $x \in B')$

$\Rightarrow x \in A'$ and $x \in B'$

$\Rightarrow x \in A' \cap B'$

$\therefore (A \cup B)' \subset A' \cap B'$ —— (1)

2) $A' \cap B' \subset (A \cup B)'$

let $x \in A' \cap B'$

$\Rightarrow x \in A'$ and $x \in B'$

$\Rightarrow x \in (U - A)$ and $x \in (U - B)$

$\Rightarrow x \in U$ and $x \in A'$ and $x \in U$ and $x \in B'$

$\Rightarrow x \in U$ and $x \in \neg(A \cup B)$

$\Rightarrow x \in (U - A \cup B)$

$\Rightarrow x \in (A \cup B)'$

$\therefore A' \cap B' \subset (A \cup B)'$ —— (2)

From (1) & (2)

$$(A \cup B)' = A' \cap B'$$

(2) Prove the following

1) $A - B = A - (A \cup B)$

We have to prove that $A - B = A - (A \cup B)$,

(i) is sufficient to prove that $A - B \subset A - (A \cap B)$ and $A - (A \cap B) \subset A - B$

(i) $A - B \subset A - (A \cap B)$

let $x \in A - B$

$\Rightarrow x \in A$ and $x \notin B$

$\Rightarrow (x \in A$ and $x \in A')$ or $(x \in A$ and $x \in B')$

$\Rightarrow x \in A$ and $(x \in A'$ or $x \in B')$

$\Rightarrow x \in A$ and $x \notin (A \cap B)$

$\Rightarrow x \in A-(A \cap B)$

$\therefore A-B \subset A-(A \cap B)$ — ①

(ii) $A-(A \cap B) \subset A-B$

let $x \in A-(A \cap B)$

$\Rightarrow x \in A$ and $x \notin (A \cap B)$

$\Rightarrow x \in A$ and $(x \in A'$ or $x \in B')$

$\Rightarrow (x \in A$ and $x \in A')$ or $(x \in A$ and $x \in B')$

$\Rightarrow x \in A$ or $x \in B'$

$\Rightarrow x \in (A-B)$

$\therefore A-(A \cap B) \subset A-B$ — ②

from ① & ②

$A-B = A-(A \cap B)$

③ Prove the Following

$(A')' = A$

Sol

Now, we prove $(A')' = A$. It is sufficient to prove that $(A')' \subset A$ and $A \subset (A')'$

(i) $(A')' \subset A$

let $x \in (A')'$

$\Rightarrow x \in U-A'$

$\Rightarrow x \in U$ and $x \notin A'$

$\Rightarrow x \in U$ and $x \in A$

$\Rightarrow x \in A$

$\therefore (A')' \subset A$ — ①

(ii) $A \subset (A')'$

$x \in A$

$\Rightarrow x \notin A'$

$\Rightarrow x \in (A')'$

$\therefore A \subset (A')'$ —— ②

From ① & ②

$(A')' = A$

Set Identities :—

Identity	Name
$A \cup \phi = A$ $A \cap U = A$	Identity Law
$A \cup U = U$ $A \cap \phi = \phi$	Domination laws
$A \cup A = A$ $A \cap A = A$	Idempotent laws
$(\bar{\bar{A}}) = A$	Complementation Law
$A \cup B = B \cup A$ $A \cap B = B \cap A$	Commutative laws
$A \cup (B \cup C) = (A \cup B) \cup C$ $A \cap (B \cap C) = (A \cap B) \cap C$	Associative Laws
$A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$ $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$	Distributive laws
$\overline{A \cup B} = \bar{A} \cap \bar{B}$ $\overline{A \cap B} = \bar{A} \cup \bar{B}$	Demorgan's laws
$A \cup (A \cap B) = A$ $A \cap (A \cup B) = A$	Absorption laws
$A \cup \bar{A} = U$ $A \cap \bar{A} = \phi$	Complement laws

1) Show that $A \cap (B-A) = \phi$

Sol: We can prove that (i) $A\cap(B-A) \subset \phi$

(ii) $\phi \subset A\cap(B-A)$

Let $x \in A\cap(B-A)$

$\Rightarrow x \in A$ and $x \in (B-A)$

$\Rightarrow x \in A$ and $x \in B$ and $x \notin A$

$\Rightarrow x \in A$ and ($x \notin A$ and $x \in B$)

$\Rightarrow$ ($x \in A$ and $x \notin A$) and $x \in B$

$\Rightarrow x \in (A-A)$ and $x \in B$

$\Rightarrow x \in \phi$ and $x \in B$

$\Rightarrow x \in \phi \cap B$

$\Rightarrow x \in \phi$

$\Rightarrow A\cap(B-A) \subset \phi$ —— (i)

Let $\phi \subset A\cap(B-A)$

As Null set is subset of Every set

$\phi \subset A\cap(B-A)$ —— (ii)

From (i) and (ii)

$\Rightarrow A\cap(B-A) = \phi$

2) Prove the Following

(i) $\overline{A\cap B} = \bar{A} \cup \bar{B}$ (ii) $\overline{A \cup (B\cap C)} = (\bar{C} \cup \bar{B}) \cap \bar{A}$

Sol: (i) $\overline{A\cap B} = \{x \mid x \notin A\cap B\}$

$= \{x \mid x \notin A \text{ or } x \notin B\}$

$= \{x \mid x \in \bar{A} \text{ or } x \in \bar{B}\}$

$= \{x \mid x \in \bar{A} \cup \bar{B}\}$

$= \bar{A} \cup \bar{B}$

$\therefore \overline{A\cap B} = \bar{A} \cup \bar{B}$

(iii) $\overline{A\cup(B\cap C)} = \bar{A}\cap(\overline{B\cap C})$

$= (\overline{B\cap C})\cap\bar{A}$ (∵ By Demorgan's law)

$= (\bar{B}\cup\bar{C})\cap\bar{A}$ (∵ By commutative law)

$= (\bar{C}\cup\bar{B})\cap\bar{A}$

Functions

Function :-

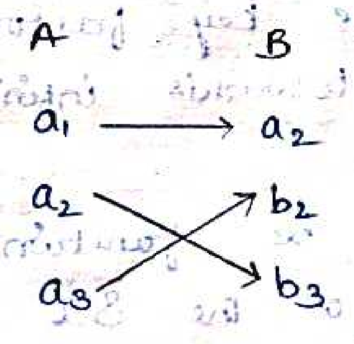
Let A and B be sets A function f from A to B is an assignment of exactly one element of B to each element of A. We write $f(a)=b$ if 'b' is the unique element of B. and assign by the function f to the element a of A. If f is a function from A to B we write $f: A \to B$

Functions are specified in many different ways. They are one - to - one or injective and onto functions.

One-to-one Function :-

A function f is said to be one-to-one or injective $\Leftrightarrow$ $f(x)=f(y)$ implies that $x=y$ $\forall$ x and y in the domain of f. A function is said to be injection if it is one-to-one.

Eg:

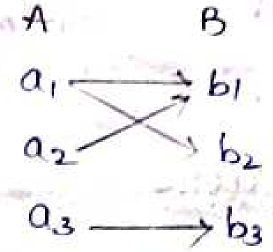


Onto Function :-

A function f from A to B is called

onto or Surjective $\Leftrightarrow$ for every element $b \in B$ there is an element $a \in A$ with $f(a) = b$

Eg :



Growth of Functions :-

The time required to solve a problem depends on more than only the number of operations it uses. The time also depends on the hardware and Software used to run the program that implemates the algorithm. How ever change hardware and Software used to implimate algorithm we can close the approximate the time required to solve a problem of size n by multiplying the previous time require by a constant. we can estimate the growth of function without quering about constant multiples or ordered terms.

Big O. Notation :-

It Seeks to describe the relative complexibility of an algorithm By reducing the growth rate to the key factors. when the key factor tends towards infinity.

Definition :-

Let f and g be functions from the Set of integers or the Set of real numbers to the set of real numbers. we say that $f(x)$ is $O(g(x))$ if there are constants C and k such that $|f(x)| \leq c|g(x)|$ whenever $x > k$. The constants c and k are witness to the relationship $f(x)$ is in order of $g(x)$

Eg:

1) Show that $f(x) = x^2+2x+1$ is $O(x^2)$

Sol: Let $f(x) = x^2+2x+1$

We observe that we can estimate the size of $f(x)$ when $x > 1$.

$$\therefore f(x) = x^2+2x+1$$
$$\leq x^2+2x^2+x^2$$
$$\leq 4x^2 \quad \forall x > 1$$

Here, $c = 4$ and $k = 1$

$$\therefore f(x) \leq 4x^2$$
$$\Rightarrow |f(x)| \leq 4|x^2|$$

from $|f(x)| \leq c|g(x)|$

$$\therefore f(x) \text{ is } O(x^2)$$

2) Show that $7x^2$ is $O(x^3)$

Sol: Given that $f(x) = 7x^2$

We know that $x^2 < x^3 \ \forall x \geq 1$

$$f(x) = 7x^2$$
$$\leq 7x^3$$
$$\therefore f(x) \text{ is } O(x^3)$$

Theorem :-

Let $f(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0$ where $a_0, a_1, a_2, \dots a_{n-1}, a_n$ are real numbers then Show that $f(x)$ is $O(x^n)$

Proof :

Since, $f(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0$

$\Rightarrow |f(x)| = |a_n x^n + a_{n-1}x^{n-1} + \cdots + a_1 x + a_0|$

$\leq |a_n|x^n + |a_{n-1}|x^{n-1} + \cdots + |a_1|x + |a_0|$

$= x^n [|a_n| + |a_{n-1}|x^{-1} + \cdots + |a_1|/x^{n-1} + |a_0|/x^n]$

$\leq x^n (|a_n| + |a_{n-1}| + \cdots + |a_1| + |a_0|)$

$\Rightarrow |f(x)| \leq Cx^n \; \forall \; x > 1$

$\therefore f(x)$ is $O(x^n)$

Eg :

1) How can Big O Notation be used to estimate the sum of the first n positive integers.

Sol

$f(x) = 1+2+3+ \cdots + n$

$\leq n+n+n+ \cdots + n$

$= n^2$

$f(x) \leq n^2$, Here $C = 1$

$\therefore f(x)$ is $O(n^2)$

2) Give Big O estimations for the factorial function and the logarithm of factorial function. where the factorial function $f(n) = n!$ is designed by $n! = 1 \cdot 2 \cdot 3 \cdots n$.

Sol

Given that

$f(x) = n!$

$f(x) = 1 \cdot 2 \cdot 3 \cdots n$

$\leq n \cdot n \cdot n \cdots n$

$= n^n$

$\therefore f(x)$ is $O(n^n)$

For logarithm :

Apply logarithm on both sides of $n! = n^n$

$\log n! \leq \log n^n$

$\Rightarrow \log n! \le n\log n$

$\therefore \log n!$ is $O(n\log n)$

The Growth of Combinations of Functions :-

Suppose $f_1(x)$ is $O(g_1(x))$ and $f_2(x)$ is $O(g_2(x))$ then $f_1(x)+f_2(x)$ is $O(\max\{g_1(x), g(x)\})$.

Eg:-

Find the Big O estimation for $f(n) = 3n\log n! + (n^2+3)\log n$ where 'n' is a positive integer.

First we consider the product $3n\log n!$

We know that $\log n!$ is the order of $n\log n$ and $3n$ is the $O(n)$ $3n\log n!$ is the $O(n^2\log n)$ — ①

Next the product $(n^2+3)\log n$

Since, n^2+3 is $O(n^2)$ when $n \geq 2$

from this $(n^2+3)\log n$ is $O(n^2\log n)$ — ②

from ① & ②

$f(n) = 3n\log n! + (n^2+3)\log n$ is the order of $n^2\log n$.

Big Omega and Big theta Notation :-

Big omega Notation :-

Let f and g be functions from the set of integers or the set of real numbers to the set of real numbers. We say that $f(x)$ is $\Omega(g(x))$ if there are positive constants C and k such that $|f(x)| \geq C|g(x)|$ whenever $x > k$.

Eg:-

Let the function $f(x) = 8x^3+5x^2+7$ is $\Omega(g(x))$

Where $g(x) = x^3$

Sol: Since, $f(x) = 8x^3+5x^2+7 \geq 8x^3$ $\forall$ positive real numbers of x. Hence, $f(x)$ is $\Omega(g(x))$

Big Theta Notation :-

Let f and g be function from the set of integers or set of real numbers to the set of real numbers. We say that $f(x)$ is $\Theta(g(x))$ if $f(x)$ is $O(g(x))$ and $f(x)$ is $\Omega(g(x))$

i.e., $C_1|g(x)| \leq |f(x)| \leq C_2|g(x)|$

Eg :-

Show that $3x^2+8x\log x$ is $\Theta(x^2)$

Sol: Since $0 \leq 8x\log x \leq 8x^2$, it follows that

$3x^2+8x\log x \leq 11x^2$ $\forall x>1$ —— ①

Consider $3x^2+8x\log x \geq 3x^2$ —— ②

From ① & ②

$\Rightarrow f(x)$ is $O(x^2)$ and $f(x)$ is $\Omega(x^2)$

Therefore $3x^2 \leq f(x) \leq 11x^2$

$\Rightarrow \Omega(x^2) \leq f(x) \leq O(x^2)$

$\therefore f(x)$ is $\Theta(x^2)$

Relations

Ordered pairs :-

Any two elements taken in a particular order enclosed in brackets and seperated by a comma are called ordered pairs.

Relation :-

let A and B are any two sets then the relation is a subset of all ordered pairs in the cartesian product. Therefore, $R \subseteq (A \times B)$

Eg: $A = (1,2,3)$, $B = \{\alpha, \beta\}$ then

$$A \times B = \{(1,\alpha), (1,\beta), (2,\alpha), (2,\beta), (3,\alpha), (3,\beta)\}$$

$$R_1 = \{(1,\alpha), (1,\beta)\}$$

$$R_2 = \{(2,\alpha), (2,\beta)\}$$

$$R_3 = \{(3,\alpha), (3,\beta)\}$$

∴ Here, R_1, R_2 and R_3 are relations

Types of Relations :-

1) Reflexive Relation :

let A be a set where a is an element of A and $(a,a) \in R$ for every element $a \in R$. Then R is called a Reflexive relation.

Eg: let $A = \{1,2,3,4\}$ and $R_1 = \{(1,1), (1,2), (1,3), (1,4), (2,1), (2,2), (2,3), (2,4), (3,1), (3,2), (3,3), (3,4), (4,4)\}$

$R_2 = \{(1,1), (1,2), (2,1), (3,1), (4,1), (4,4)\}$

$R_3 = \{(4,1), (1,2), (2,1), (2,2), (3,1), (3,3), (4,1), (4,2), (4,$

Here R_1, R_3 are Reflexive.

2) Irreflexive Relation :

The relation R on the set A is

irreflexive. If $(a,a) \notin R$ for every element $a \in A$.

3) Symmetric relation :

Let $a,b \in A$ then if $(a,b) \in R \Rightarrow (b,a) \in R$ then the relation R is said to be symmetric

Eg :- $A = \{1,2,3\}$, $R_1 = \{(1,1),(1,2),(2,1)\}$

Here, R is a symmetric relation

4) Anti-symmetric Relation :-

Let $a,b \in A$ if $(a,b) \in R$ and $(b,a) \in R$ only if $a = b$ $\forall a,b \in A$

Eg: $A = \{1,2,3,4\}$ and $R_1 = \{(1,1),(1,2),(1,3),(1,4),(2,2),(2,3),(2,4),(3,3),(3,4)\}$

Here R_1 is Antisymmetric

5) Transitive Relation :-

Let $a,b,c \in A$ if $(a,b) \in R$ and $(b,c) \in R$ then $(a,c) \in R$. then R is said to be a transitive relation.

Eg: $A = \{1,2,3\}$, $R_1 = \{(1,2),(2,3),(1,3),(3,1),(2,1),(1,1)\}$

(6) Equivalence Relation :-

A relation R is said to be equivalence relation if R is reflexive, Symmetric and transitive

7) Combining Relations :-

The relations can be combined by using set operators $\cup, \cap, -$

Eg :- $A = \{1,2,3\}$, $R_1 = \{(1,1),(2,2),(3,3)\}$, $R_2 = \{(1,1),(1,2),(1,3),(1,4)\}$ then

$R_1 \cup R_2 = \{(1,1),(1,2),(1,3),(1,4),(2,2),(3,3)\}$

$R_1 \cap R_2 = \{(1,1)\}$

$R_1 - R_2 = \{(2,2),(3,3)\}$

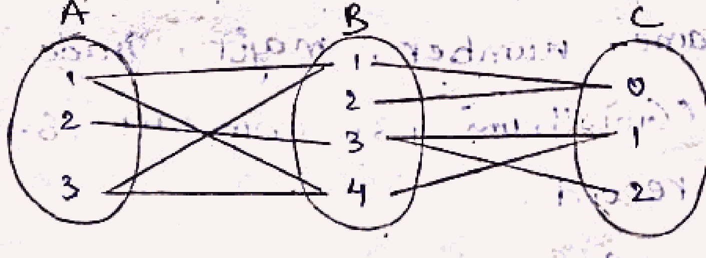
8) Composite Relation :-

Let R be a relation from a set A to a set B and S is a relation from a set B to a set C. The composite relation of R and S is the relation consists of ordered pairs (a,c) where $a \in A$ and $c \in C$. And for which there exists an element $b \in B$ such that $(a,b) \in R$ and $(b,c) \in S$. Then it is called a composite relation of R and S which is denoted by SOR.

Eg :-

What is the composite relation of R and S where R is a relation from {1,2,3} to {1,2,3,4} with R = {(1,1), (1,4), (2,3), (3,1), (3,4)} and S is the relation from {1,2,3,4} to {0,1,2} with S = {(1,0), (2,0), (3,1), (3,2), (4,1)}

Sol



SOR = {(1,0), (1,1), (2,1), (2,2), (3,0), (3,1)}

n-ary Relation :-

Let $A_1, A_2 \ldots A_n$ be sets. Any n-ary relation on these sets is a subset of $A_1 \times A_2 \times \ldots \times A_n$. These sets $A_1, A_2, \ldots A_n$ are called the domains of the relation and n is called the degree.

Eg:-

let R be a Relation on $N \times N \times N$ consisting of triple (a, b, c) where $a < b < c$ then $(1, 2, 3) \in R$ and $(2, 4, 3) \notin R$

Here the degree of relation R is '3'.

Database and Relations :-

The relational data model based on the concept of a relation. A database consists of records which are n-tuples made of fields; the fields are the entries of n-tuples. For example a database of student records may be made up of fields containing name, number, major and gradepoint. The relational data model represents a database of records as an n-ary relation. The student records are represented as tuples of the form (name, number, major, grade point). For example (Gowthami, 13, Computer, 8.1) is a sample record.

Relations used to represent databases are also called tables.

Primary Key :-

A domain of an n-ary relation is called a primary key when the value of the n-tuple from this domain determines the n-tuple i.e., a domain is a primary key when no two n-tuples in the relation have the same value from this domain.

Eg:-

Student name	ID NO	Major	Grade point
Gowthami	208021024	CS	8.1
Malleswari	208021029	CS	8.1
Tejasri	208021030	CS	7.9

In the above table ID No is the primary key. Since, it has distinct values in its column

Composite key :-

Combinations of domains can also uniquely identify n-tuples in an n-ary relation. The cartesian product of those domains is called a composite key.

Operations on n-ary relation :-

1) **Selection :**

let R be n-ary relation and c is a condition that elements in R satisfied then the selection operator 'S_c' maps n-ary relation R to the n-ary relation of all n-tuples from R that satisfy the condition c.

2) **Projection :**

The projection $P_{i_1, i_2, \ldots i_m}$ maps n-tuple $(a_1, a_2, \ldots, a_n)$ to multiple $(a_{i_1}, a_{i_2}, \ldots a_{i_m})$ where $m \le n$

3) **Join :**

The join operation is used to combine two tables into one when these tables share some identical fields.

Representing Relations :-

The Relations can be represented in many ways. one of the way is to list ordered pairs. There are two alternatives for representing relations those are zero-one matrices and directed graphs.

Representing relations using Matrices :-

A relation between finite sets can be represented using a zero-one matrix. Suppose, R is a relation from $A = \{a_1, a_2 \cdots a_m\}$ to $B = \{b_1, b_2 \cdots b_n\}$. The relation R can be represented by the Matrix $M_R = [m_{ij}]$ where $m_{ij} = \begin{cases} 1 & \text{if } (a_i, b_j) \in R \\ 0 & \text{if } (a_i, b_j) \notin R \end{cases}$

Eg:-

Suppose $A = \{1, 2, 3\}$, $B = \{1, 2\}$. Let R be a relation from A to B, containing $(a, b) \in R$ if $a > b$, then

$A \times B = \{(1,1), (1,2), (2,1), (2,2), (3,1), (3,2)\}$

$R = \{(2,1), (3,1), (3,2)\}$

It is represented by a matrix

$$M_R = \begin{array}{c c} & \begin{array}{cc} 1 & 2 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \end{array} & \begin{bmatrix} 0 & 0 \\ 1 & 1 \\ 1 & 1 \end{bmatrix} \end{array}$$

The zero-one matrix for a reflexive relation:-

The relation R is reflexive if and only if $m_{ii} = 1$ for $i = 1, 2, \ldots n$. In other words R is reflexive all the elements in the matrix diagonal of M_R are 1's

The zero-one matrix for a Symmetric Relation :-

The Relation R is Symmetric if and only if $(a,b) \in R \Rightarrow (a,b) \in R$. In terms of the entries of M_R, R is Symmetric if and only if $m_{ji} = 1$ whenever $m_{ij} = 1$

The zero-one matrix for an anti-symmetric relation :-

The relation R is anti symmetric if and only if $(a,b) \in R$ and $(b,a) \in R \Rightarrow a = b$. In terms of the entries of M_R if $m_{ij} = 1$ with $i \neq j$ then $m_{ji} = 0$ (or) in other words $m_{ij} = 0$ (or) $m_{ji} = 0$ where $i \neq j$

Eg :-

Suppose the relation R of on a set is represented by the matrix $M_R = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix}$

R is reflexive, symmetric and not anti-symmetric.

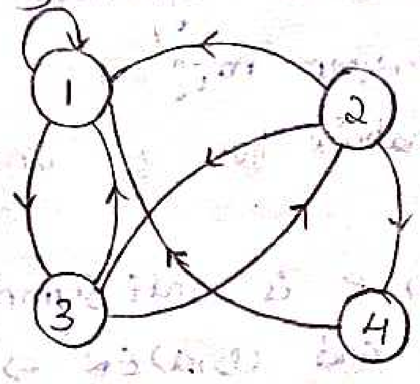
Representing relations using directed graph :-

A directed graph or digraph consists of a set V of vertices or nodes together with the edges set E of ordered pairs of elements of V. called edges or arc's. The vertex a is called the initial vertex of the edge (a,b) and the vertex b is called the terminal vertex of this edge

An Edge of the form (a,a) is represented using an arc from the vertex a to itself. Such an edge is called loop

Eg:

1) Draw the directed graph of the relation $R = \{(1,1), (1,3), (2,1), (2,3), (2,4), (3,1), (3,2), (4,1)\}$ on the set $\{1,2,3,4\}$



2) What are the ordered pairs

Closure of Relations :-

1) Reflexive Closure :

The Relation $R = \{(1,1), (1,2), (2,1), (3,2)\}$ on the set $A = \{1,2,3\}$ is not Reflexive. we can produce a Reflexive relation containing R as small as possible by adding (2,2) and (3,3) to R. Since these are the only pairs of the form (a,a) that are not in R. clearly this new relation contains R and any Reflexive relation that contains R must also contain (2,2) and (3,3). Because this relation contains R, is reflexive and it is contained within every Reflexive relation that contains R, it is called Reflexive closure of R

Eg:

1) what is the Reflexive closure of relation $R = \{(a,b)$ such that $a<b\}$ on the set of integers

Sol The reflexive closure of R is $R \cup \Delta$ = $\{(a,b) / a<b\} \cup \{(a,a) / a \in Z\} = \{(a,b) / a \leq b\}$

2) Symmetric closure :

The Relation $R = \{(1,1),(1,2),(2,2),(2,3),(3,1),(3,2)\}$ on the set $A = \{(1,2,3)\}$ is not symmetric. We can produce a symmetric relation i.e., as small as possible and contains R by adding (2,1) and (1,3). Since these are the only pairs of the form (b,a) with (a,b) belongs to R that are not in R. This new relation is symmetric and contains R. This new relation is called Symmetric closure of R.

Eg:

1) what is the symmetric closure of the Relation $R = \{(a,b) / a>b\}$ on the set of +ve integers

Sol The symmetric closure of R is

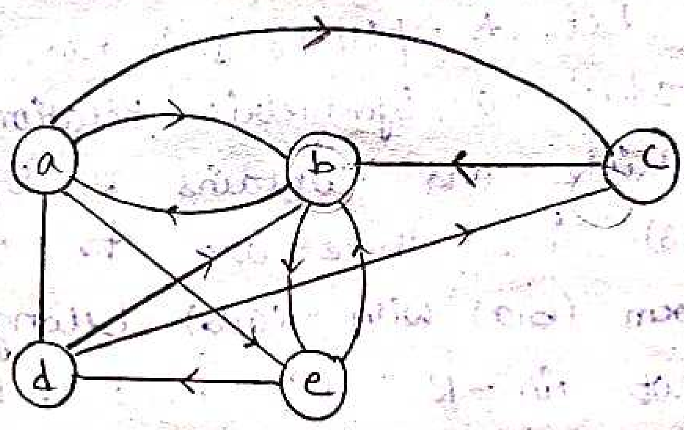
$R \cup R^{-1} = \{(a,b) / a>b\} \cup \{(b,a) / a>b\}$

$= \{(a,b),(b,a) / a \neq b\}$

Paths in diagrams :-

A path from a to b in the diagram is a sequence of edges $(x_0, x_1), (x_1, x_2) \cdots\cdots (x_{n-1}, x_n)$ in G where n is a non-negative integer and $x_0 = a$, and $x_n = b$, i.e., the Sequence of edges where the terminal vertex of an edge is same as the initial vertex in the next edge in the path. This path is denoted by $x_0, x_1, \ldots x_n$ and has length n. A path of length $n \geq 1$ that begins and ends at the same vertex is called a circuit or a cycle.

Eg: Find various paths in the directed graph shown below.



Path 1 : a, b, e, d

length of the path 3

Path 2 : a, e, d, c, b

length of the path 4

3) Transitive closure :

Connectivity Relation :

let R be a Relation on a set A. The connectivity relation R^* consists of the pairs (a,b) such that there is a path of length atleast one from a to b in R. In other words $R^* = \bigcup_{n=1}^{\infty} R^n$. i.e., R^n consists of the pairs (a,b) such that there is a path of length n from a to b it follows that R^* is the union of all the sets R^n.

Eg: Let R be a Relation on a Set A. A is said to be all people in the world that contains (a,b) if a has met b what is R^n where n is position integer greater than one what is R^*.

Sol: The Relation R^2 contains (a,b) if there is a person c such that $(a,c) \in R$ and $(c,b) \in R$ i.e., if their is a person c such that a has met c and c has met b. Similarly R^n consists of those pairs (a,b) such that there are people $x_1, x_2 \ldots x_{n-1}$ such that a has met x_1, x_1 has met x_2 and so on x_{n-1} as met b. The Relation R^* contains (a,b) if there is a sequence of people starting with a and ending with b such that each person in the sequence as met the next person in the sequence.

Theorem :-

The transitive closure of a Relation R equals the connectivity relation R^*.

Proof :

Note that R^* contains R by definition. To show that R^* is the transitive closure of R. we must also show that R^* is transitive and $R^* \subseteq S$ whenever S is a transitive relation that contains R.

First we show that R^* is transitive. If ~~a and~~ $(a,b) \in R^*$ and $(b,c) \in R^*$, then there are paths from a to b and from b to c in R. we can obtain a path from a to c by starting with the path from a to c and following it with the path b to c. Hence $(a,c) \in R^*$, it follows that R^* is transitive.

Now suppose that S is a transitive relation that contains R. Since S is transitive then S^n is also transitive. For the more $S^* = \bigcup_{k=1}^{\infty} S^k$ and $S^k \subseteq S$ it follows that $S^* \subseteq S$. Note that if $R \subseteq S$ then $R^* \subseteq S^*$. Consequently $R^* \subseteq S^* \subseteq S$. Thus only transitive relation that contains R must also contains R^*. Therefore R^* is the transitive closure of R.

Warshall's Algorithm :

Warshall's Algorithm is based on the construction of a sequence of 0-1 matric. These matrices are $W_0, W_1, \ldots W_n$ where $W_0 = M_R$ is the 0-1 matrix of this relation $W_K = [W_{ij}(k)]$ where $W_{ij}(k) = 1$ if there is a path from V_i to V_j such that all the interior (mediate) vertices of this path are in the set $\{V_1, V_2 \ldots V_k\}$ and is 0 otherwise.

Procedure : Procedure Warshall (M_R : n×n zero-one matrix)

$W := M_R$

for $k := 1$ to n

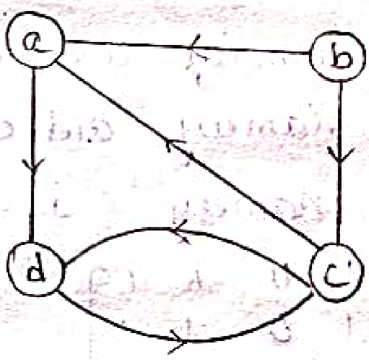
begin

for $i = 1$ to n

```
begin
    for j := 1 to n
        w_ij = w_ij ∨ (w_ik ∧ w_kj)
    end
end { W = [w_ij] is M_R* }
```

Eg:

let R be the relation with directed graph shown in the following figure a,b,c,d be a distinct elements of the set. find the matrices W_0, W_1, W_2, W_3 and W_4. The matrix W_4 is the transitive closure of R



let $v_1 = a, v_2 = b, v_3 = c$ and $v_4 = d$. W_0 is the matrix of the relation. Hence

$$W_0 = \begin{array}{c} \\ a \\ b \\ c \\ d \end{array} \begin{array}{c} \begin{array}{cccc} a & b & c & d \end{array} \\ \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \end{array}$$

W_1 has 1 as its $(i,j)^{th}$ entry

There are intermediate vertices from b to d namely b,a,d. Hence

$$W_1 = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

W_2 is same as W_1. Hence the matrix same as W_1 as no intermediate matrices are there from b.

W_3 we have paths from d to a, namely d,c,a and from d to d namely d,c,d. Hence

$$W_3 = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 \end{bmatrix}$$

Finally W_4 has 1 as its $(i,j)^{th}$ entry if there is a path from v_i to v_j we have following paths.

(i) a to a namely a,d,c,a

(ii) a to c namely a,d,c

(iii) c to c namely c,d,c

$$W_4 = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 \end{bmatrix}$$

$\therefore$ W_4 is the transitive closure of the given relation R.

<u>Equivalence</u> <u>Relation</u> :-

A Relational on a set A is called an Equivalence Relation. If it is reflexive, symmetric and transitive.

<u>Eg</u>!

1) Suppose the R is a relation on the set of strings of English letters such that aRb $\Leftrightarrow$ $l(a) = l(b)$ where $l(x)$ is the length of the string x. is R an Equivalence Relation.

Sol: We have to verify the relation R is a Equivalence Relation, we have to verify whealther R holds reflexive, Symmetric and transitive properties.

Since $l(a) = l(b)$ is given in The problem i.e. we can say $a = a$ i.e. $l(a) = l(a)$. It follows that aRa such that $(a,a) \in R$ whenever a is string. So that R is Reflexive.

Suppose aRb then $l(a) = l(b)$ it follows bRa since $l(b) = l(a)$ Hence R is Symmetric

Suppose that aRb and bRc then $l(a) = l(b)$ and $l(b) = l(c)$ hence $l(a) = l(c)$. So that R is transitive. Since R is reflexive, Symmetric and transitive It is an Equivalence Relation.

2) Let R be the Relation on the set of real numbers such that $aRb \Leftrightarrow a-b$ is an Integer is R an Equivalence Relation.

Sol: Since, $a-a=0$ is an Integer for all real number a, aRa for all real numbers a, hence R is an Reflexive. Now, Suppose that aRb then $a-b$ is an integer so that $b-a$ is also integer. Hence bRa it follows that R is Symmetric.

If aRb and bRc then $a-b$ and $b-c$ are Integers $a-c = (a-b)+(b-c)$ is an integer. hence aRc, thus R is transitive

Consequently R is an equivalence relation

Equivalence Classes :-

Let R be a equivalence be a equivalence

relation on a set A. The set of all elements that are related to an element $a \in A$ is called the equivalence class of a. The equivalence class of a with respect to R is denoted by $[a]_R$

When only one relation is under consideration we will delete the subscript R and write $[a]$ for this equivalence class. In other words the equivalence class of the element a is $[a]_R = \{s / (a,s) \in R\}$.

If $b \in [a]_R$ then b is called as representative of this equivalence class.

Eg:

1) What is the equivalence class of an integer for the equivalence relation R i.e., $aRb \Leftrightarrow a = b$ (or) $a = -b$.

Sol:-

Since, an integer is equivalent to itself and its negative in this equivalence relation it follows that $[a]_R = \{a, -a\}$.

The set contains two distinct integers unless $a = 0$ i.e., $[a]_R = \{b, -b \Rightarrow a \neq b\}$

*** Theorem :- 1.

Let R be an equivalence relation on a set A. So that the following statements are equivalent.

(i) aRb (ii) $[a] = [b]$ (iii) $[a] \cap [b] \neq \phi$

Proof:

First we show that (i) implies (ii) Assume that aRb we will prove that $[a]=[b]$. By showing $[a]\subseteq[b]$ and $[b]\subseteq[a]$.

Suppose $c\in[a]$ then aRc since aRb and R is symmetric. we know that bRa.

Since bRa, aRc and R is transitive then bRc / $c\in[b]$.

Therefore $[a]\subseteq[b]$ —— ①

let $c\in[b] \Rightarrow bRc$,

Since aRb and bRc and R is transitive then aRc such that $c\in[a]$

Therefore $[b]\subseteq[a]$ —— ②.

From ① & ② $\Rightarrow [a]=[b]$

We will show that

(ii) implies (iii)

Assume that $[a]=[b]$, it follows that $[a]\cap[b]\neq\phi$. since $[a]$ is non-Empty. (therefore $a\in[a]$ because R is reflexive).

We will show that (iii) $\Rightarrow$ (i)

Suppose that $[a]\cap[b]\neq\phi$, then there is an element $c\in[a]$ and $c\in[b]$, in other words, aRc and bRc by the symmetric property of R, cRb then by transitive property since, aRc and cRb we have aRb

Since, (i) $\Rightarrow$ (ii), (ii) $\Rightarrow$ (iii) and (iii) $\Rightarrow$ (i). The statements (i), (ii) and (iii) are equivalent.

Partial Orderings :- xxx Imp topic

A relation R on a set S is called a partial ordering (or) partial order if it is reflexive, anti-symmetric and transitive. A set S together with a partial ordering R is called as partially ordered set or poset and is denoted by (S,R).

Partition :-

A partition of a set S is a collection of disjoint non-empty subsets of S. That have S there as a union. In other words, the collection of subset A_i, $i \in I$ forms a partition of S. If and only if $A_i \neq \phi$ for $i \in I$, $A_i \cap A_s = \phi$ when $i \neq s$ and $\bigcup_{i \in I} A_i = S$

Eg:

Suppose that $S = \{1,2,3,4,5,6\}$ then the collection of sets $A_1 = \{1,2,3\}$, $A_2 = \{4,5\}$ and $A_3 = \{6\}$ forms a partition of S, since these sets are disjoint and there union is S.

Eg :-

Show that "the greater or equal ($\geq$)" relation is a partial ordering on the set of integers.

Sol: Since $a \geq a$ for every integer a. Hence, "$\geq$" is reflexive. If $a \geq b$ and $b \geq a$ then "$\geq$" is anti-symmetric

Since $a \geq b$ and $b \geq c$ then such that $a \geq c$ hence, "$\geq$" is transitive.

As "$\geq$" holds reflexive, anti-symmetric and transitive. It is a partial ordering on the set of integers.

Comparable and Incomparable elements :-

The elements a and b of a poset $(S, \leq)$ are called comparable if either $a \leq b$ (or) $b \leq a$

When a and b elements of S such that neither $a \leq b$ nor $b \leq a$; a and b are called Incomparable

Eg :-

In the poset $(Z^+, /)$ are the integers 3 and 9 comparable are 5 and 7 comparable?

Sol:- The integers 3 and 9 are comparable since $3/9$.

The integers 5 and 7 are incomparable because $5 \nmid 7$ and $7 \nmid 5$

Total Ordering :-

If $(S, \leq)$ is a poset and every two elements of S R are comparable S is called totally ordered or linearly ordered set and $\leq$ is called total ordered or linear ordered. A totally ordered set is also called as chained.

Eg:

The poset $(Z, \leq)$ is a to totally ordered set since $a \leq b$ or $b \leq a$ when a and b are integers.

Well - Ordered Set :-

$(S, \leq)$ is a well ordered set if it is a poset such that, $\leq$ is a total ordering and such that every non-empty subset of S has a least element.

Lexico graphic Ordering :-

The lexico graphic ordering $<$, on $A_1 \times A_2$ is defined by specifying that one pair is less than second pair if the first entry of the first pair is less than the first entry of the second pair or if the first entries are equal, but the second entry of pair. It is less than the second entry of second pair.

Eg :

Determine whether $(3,5) < (4,8)$ whealther $(4,9) < (4,11)$ in the poset $(Z \times Z, \leq)$ where $\leq$ is the lexico graphic ordering which is constructed from the usual $\leq$ relation on Z.

Sol: (i) $(3,5) < (4,8)$

Since, $3 < 4, 5 < 8$

(ii) $(4,9) < (4,11)$

Since, the first terms in both the pairs are equal i.e., $4=4$. verify the second elements

But $9 < 11$.

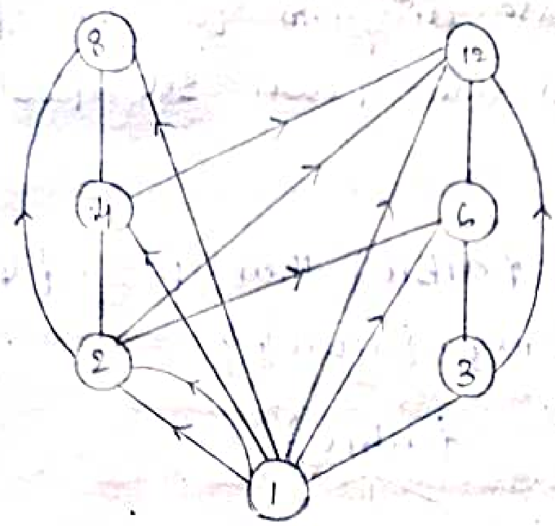
Hasse diagrams :-

These diagrams are used to represent the relation between the elements in the poset.

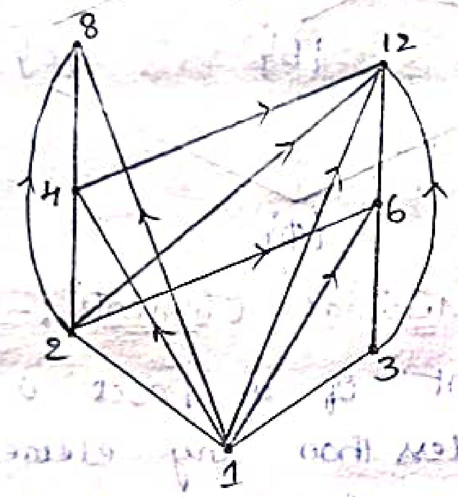
Eg :-

1) Draw the Hase diagram representing the partial ordering $\{(a,b) / a \text{ divides } b\}$ on $\{1,2,3,4,6,8,12\}$

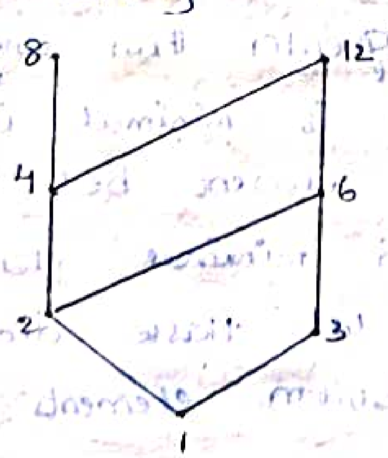
Sol: Begin with the digraph for this partial order as shown in the following figure.



Now, removing all the loops as shown in the following Figure.



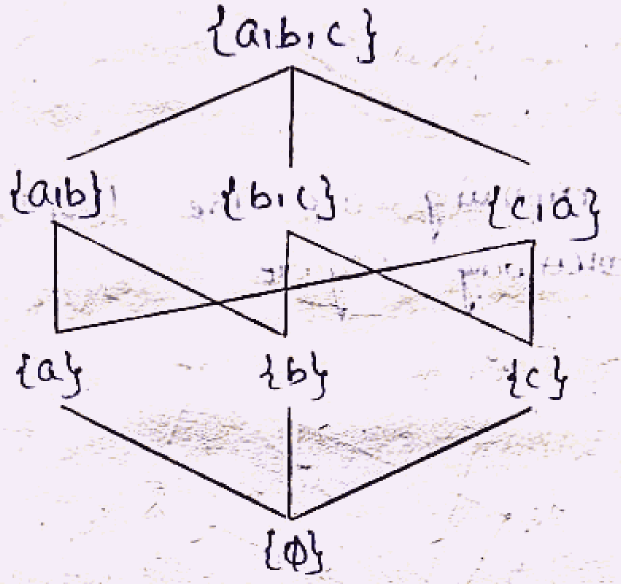
The delete all the edges implied by the transitive property. These are {(1,4),(4,6),(1,8),(1,12), (2,8),(2,12) and (3,12)}



Arrange all edges to point upward and delete all arrows to obtain the Hasse diagram.

2) Draw the Hasse diagram for the partial ordering $\{(A,B) / A \subseteq B\}$ on the power set $P(S)$ where $S = \{a,b,c\}$.

Sol:- Since, $S = \{a,b,c\}$ then $P(S) = \{\phi, \{a\}, \{b\}, \{c\}, \{a,b\}, \{b,c\}, \{c,a\}, \{a,b,c\}\}$



<u>Maximal and Minimal Elements</u> :-

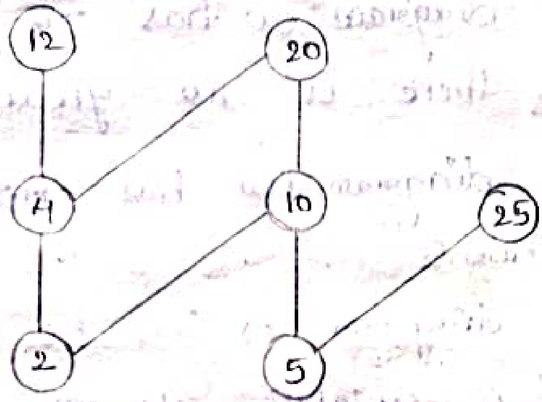
An element of a poset a called maximal if it is not less than any element of the poset i.e., a is maximal in the poset $(S, \leq)$ if there is no $b \in S$ such that $a < b$.

An Element of a poset is called minimal if it is not greater than any element of the poset i.e, a is minimal in the poset $(S, \leq)$ if there is no element $b \in S$ such that $b < a$

Maximal and minimal elements are easy to spot using 'a'. Hasse diagram they are the top and bottom elements in the diagram

Eg:-

Which elements of the poset $(\{2,4,5,10,12,20,25\}, /)$ represented by using hasse diagram, are maximal or which are minimal.



Sol: In the above Hasse diagram,

The maximal elements are 12, 20, 25

The minimal elements are 2, 5

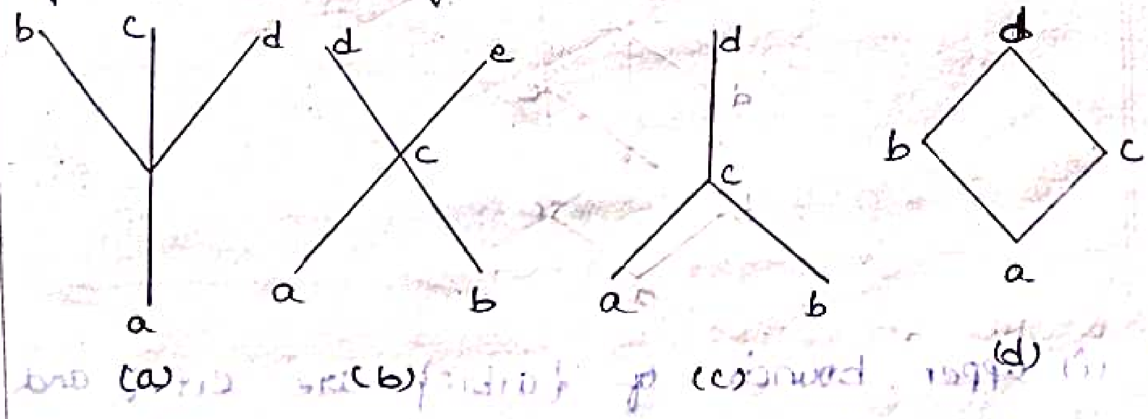
Least and greatest Elements :-

An element in a poset that is greater than every other element is called the greatest element i.e., a is the greater element of the poset $(S, \leq)$ if $b \leq a \ \forall a \in S$.

An Element is called the least element if it is less than all other elements in the poset i.e., a is least elements of the poset $(S, \leq)$ if $a \leq b \ \forall b \in S$

Eg:

Determine whether the poset represented by each of the Hasse diagram in the following figure have a greater and least element.



Sol: From the above Hasse diagram

Hasse diagram (a) has one least element i.e., a, but there is no greatest element.

Hasse diagram (b) has neither least nor greatest elements.

Hasse diagram (c) has no least element but one (d) is greatest element.

Hasse diagram (d) has no least element i.e., a, and one greatest element i.e., d

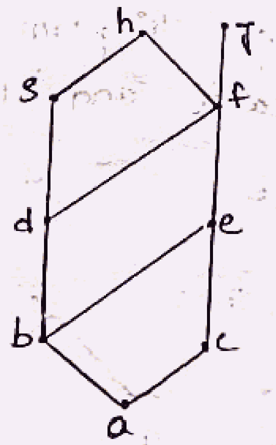
Upper bound and Lower bound :-

A Subset A of a poset $(S, \leq)$ if U is an element of S such that $a \leq U$ for all elements $a \in A$, then U is called an upper bound of A

Likewise if l is an element of S such that $l \leq a$ for all elements $a \in A$, then l is called a lower bound of A.

Eg:-

Find the lower and upper bounds of the subsets $\{a,b,c\}$, $\{J,h\}$ and $\{a,c,d,f\}$ in the poset with the Hasse diagram shown below.



Sol (i) upper bounds of $\{a,b,c\}$ are e, f, J and h
lower bound of $\{a,b,c\}$ is a

(ii) There are no upper bounds for $\{J,h\}$

Lower bounds of {J,h} are a,b,c,d,e,f

(iii) upper bounds of {a,c,d,f} are J and h

lower bounds of {a,c,d,f} is a

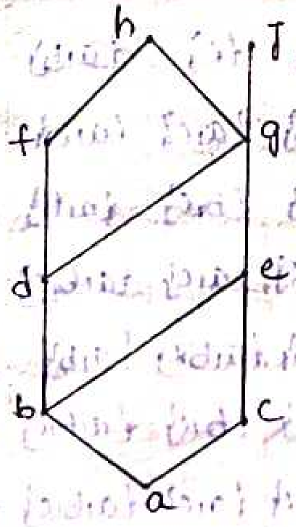
Least upper bound and greatest lower bound :-

The element x is called the least upper bound of subset A if x is an upper bound that is less than every other upper bound A.

Similarly, the element y is called greatest lower bound of A if y is the lower bound of A and $z \leq y$ whenever 'z' is a lower bound of A then the element y is called the greatest lower bound.

Eg:

(1) Find the greatest lower bound and the least upper bound of {b,d,g} if they exist in the poset shown in the following figure.



Sol: The upper bound of {b,d,g} are J and h since, $J < h$, the least upper bound is 'g'

The lower bound of {b,d,g} are a and b since $a < b$ the greatest lower bound is 'b'

(2) Write greatest lower bound and least upper

bound table for the Power set of A, for the set $A = \{a,b,c\}$

Sol: $A = \{a,b,c\}$

$P(A) = \{\phi, \{a\}, \{b\}, \{c\}, \{a,b\}, \{b,c\}, \{c,a\}, \{a,b,c\}\}$

GLB $\cap^*$	ϕ	$\{a\}$	$\{b\}$	$\{c\}$	$\{a,b\}$	$\{b,c\}$	$\{a,c\}$	$\{a,b,c\}$
ϕ	ϕ	ϕ	ϕ	ϕ	ϕ	ϕ	ϕ	ϕ
$\{a\}$	ϕ	$\{a\}$	ϕ	ϕ	$\{a\}$	ϕ	$\{a\}$	$\{a\}$
$\{b\}$	ϕ	ϕ	$\{b\}$	ϕ	$\{b\}$	$\{b\}$	ϕ	$\{b\}$
$\{c\}$	ϕ	ϕ	ϕ	$\{c\}$	ϕ	$\{c\}$	$\{c\}$	$\{c\}$
$\{a,b\}$	ϕ	$\{a\}$	$\{b\}$	ϕ	$\{a,b\}$	$\{b\}$	$\{a\}$	$\{a,b\}$
$\{b,c\}$	ϕ	ϕ	$\{b\}$	$\{c\}$	$\{b\}$	$\{b,c\}$	$\{c\}$	$\{b,c\}$
$\{c,a\}$	ϕ	$\{a\}$	ϕ	$\{c\}$	$\{a\}$	$\{c\}$	$\{c,a\}$	$\{c,a\}$
$\{a,b,c\}$	ϕ	$\{a\}$	$\{b\}$	$\{c\}$	$\{a,b\}$	$\{b,c\}$	$\{c,a\}$	$\{a,b,c\}$

LUB $\cup^*$	ϕ	$\{a\}$	$\{b\}$	$\{c\}$	$\{a,b\}$	$\{b,c\}$	$\{a,c\}$	$\{a,b,c\}$
ϕ	ϕ	$\{a\}$	$\{b\}$	$\{c\}$	$\{a,b\}$	$\{b,c\}$	$\{a,c\}$	$\{a,b,c\}$
$\{a\}$	$\{a\}$	$\{a\}$	$\{a,b\}$	$\{a,c\}$	$\{a,b\}$	$\{a,b,c\}$	$\{a,c\}$	$\{a,b,c\}$
$\{b\}$	$\{b\}$	$\{a,b\}$	$\{a,b\}$	$\{b,c\}$	$\{a,b\}$	$\{b,c\}$	$\{a,b,c\}$	$\{a,b,c\}$
$\{c\}$	$\{c\}$	$\{a,c\}$	$\{b,c\}$	$\{a,c\}$	$\{a,b,c\}$	$\{b,c\}$	$\{a,b,c\}$	$\{a,b,c\}$
$\{a,b\}$	$\{a,b\}$	$\{a,b\}$	$\{a,b\}$	$\{a,b,c\}$	$\{a,b\}$	$\{a,b,c\}$	$\{a,b,c\}$	$\{a,b,c\}$
$\{b,c\}$	$\{b,c\}$	$\{a,b,c\}$	$\{b,c\}$	$\{b,c\}$	$\{a,b,c\}$	$\{b,c\}$	$\{a,b,c\}$	$\{a,b,c\}$
$\{c,a\}$	$\{a,c\}$	$\{a,c\}$	$\{a,b,c\}$	$\{a,c\}$	$\{a,b,c\}$	$\{a,b,c\}$	$\{a,c\}$	$\{a,b,c\}$
$\{a,b,c\}$	$\{a,b,c\}$	$\{a,b,c\}$	$\{a,b,c\}$	$\{a,b,c\}$	$\{a,b,c\}$	$\{a,b,c\}$	$\{a,b,c\}$	$\{a,b,c\}$

Lattice :-

Let p be a non-empty set and $(P, \leq)$ is a poset. The poset with infeumment (LUB and

GLB) is called lattice. i.e, The lattice is a poset in which any two elements have infimum and supremum.

(or)

A partially ordered set in which every pair of elements has both a least upper bound and a greatest lower bound is called a lattice.

Sub Lattice :-

Let $(L, \wedge, \vee)$ is a lattice for any non-empty subset S of L is said to be sub lattice it satisfies $a \vee b, a \wedge b \in L, S$.

Eg :-

Determine whether the posets $(\{1,2,3,4,5\}, /)$ and $(\{1,2,4,8,16\}, /)$ are lattices.

Sol: (i) for the poset $(\{1,2,3,4,5\}, /)$, it is not a lattice. Since, there are no upper bounds for 2, 3.

(ii) For the poset $(\{1,2,4,8,16\}, /)$ every two elements of the second poset have both a least upper bound and greatest lower bound. Hence, it is a lattice.

Distributive lattice :-

A system $(L, \vee, \wedge)$ is said to be distributive lattice if 'L' is a lattice and it is satisfying distributive property i.e, if any $a, b, c \in L$,

$$a \vee (b \wedge c) = (a \vee b) \wedge (a \vee c)$$

$$a \wedge (b \vee c) = (a \wedge b) \vee (a \wedge c).$$

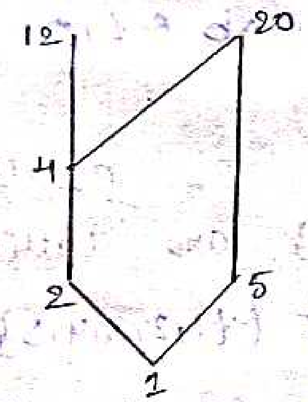
Topological Sorting :-

A total ordering $\leq$ is said to be comparable with the partial ordered R if $a \leq b$ whenever aRb constructing a comparable total ordering from a partial ordering is called topological sorting.

Eg:

Find a comparable total ordering from the poset $(\{1,2,4,5,12,20\}, /)$.

Sol: Draw the Hasse diagram for the given poset.



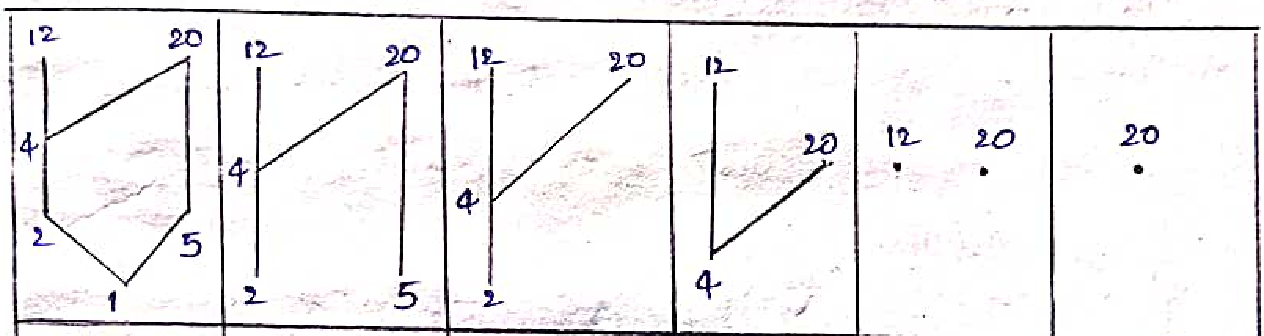
Step-1: Choose a minimal element. This must be 1, since it is the only minimal element

Step-2: Select a minimal element of $(\{2,4,5,12,20\}, /)$ there are two minimal elements in the poset namely 2 and 5. we select 5 the remaining elements are $\{2,4,12,20\}$ The only minimal element of this stage is 2.

Step-3: Select a minimal element of $(\{4,12,20\}, /)$ This must be 4. Since, it is the only minimal element.

Step-4: Now, we have only two elements namely 12 and 20 both are minimal elements of ({12,20}, 1) either can be choosen in this as minimal element. Select 12 has the minimal element which leaves 20 has the last element

The above steps produces the total ordering $1 \prec 5 \prec 2 \prec 4 \prec 12 \prec 20$. The steps used by this sorting algorithm is displaying with following diagram.



Minimal elements choosen 1	5	2	4	12	20

18/12/13 Permutations And Combinations

* Product Rule:-

Suppose that a procedure can be broken down into a sequence of two tasks. If there are n_1 no.of ways to do the first task and n_2 no.of ways to do the second task after the first task has been done. Then there are $n_1 \cdot n_2$ ways to do the procedure.

eg:- 1) A new company with just two employees Sanchez and Patel, rents a floor of a building with 12 officers. How many ways are there to assign different officers to these two employees?

Sol. The procedure of assigning officers to these two employees can be done as 12 ways for the first employee then assigning an office to the second person different from the office assigned to the first person, which can be done in 11 ways.

∴ By the product rule there are $12 \times 11 = 132$ ways to assign offices to these employees.

2) The chairs of an auditorium are to be labelled with a letter and a +ve integer not exceeding 100 what is the largest no.of chairs that can be labelled differently?

Sol. The procedure of labelling a chair consists of two tasks, namely assigning one of the 26 letters and then assigning one of the 100 posible integers to the seat. By the product rule there are $26 \times 100 = 2600$ different ways that a chair can be labelled.

3) There are 32 micro computers in a computer center. Each micro computer has 24 ports. How many different ports to a micro computer in the centre are there?

Sol. The procedure of choosing a port consists of two tasks, first picking, a micro computer and then picking a port on that computer. Because there are 32 ways to choose the micro computer and 24 ways to choose the port.

By the product rule there are $32 \times 24 = 768$ different ports in the computer center.

4) How many different bit strings of length 7 are there?

Sol. Each of the 7 bits can be choosen in 2 ways, because is bit is 0 (or) 1. By the product rule there are a total of $2^7 = 128$, Different bit strings of length 7.

5) How many different liscence plates are available if each plate contains a sequence of 3 letters followed by 3 digits?

Sol: There are 26 choices for each of the three letters and 10 choices for each of the 3 digits. Hence by the product rule there are a total of $26 \times 26 \times 26 \times 10 \times 10 \times 10 = 17576000$ possible liscence plates.

* Sum Rule:-

If a task can be done either in one of n_1 ways or in one of n_2 ways, where non of the set of n_1 ways is the same as any of the set of n_2 ways, then there are (n_1+n_2) ways to do the task.

eg:- 1) suppose that either a member of the mathematics faculty (or) a student who is a mathematics major is choosen as a representative to a university commitee.

tative if there 37 members of the mathematics faculty and 83 mathematics majors and no one is both a faculty and a student.

Sol There are 37 ways to choose a member of the mathematics faculty and there are 83 ways to choose a student who is a mathematics major. Since no one is both faculty and student.

By the sum rule there are $37+83 = 120$ possible ways to pick this representative.

2) A student can choose a computer project from one of three lists. The 3 lists contain 23, 15, 19 possible projects respectively. No project is in more than one list. Howmany possible projects are there to choose from?

Sol. The student can choose a project from the first list, second list (or) the third list, because no project is in more than one list.

By the sum rule there are $23+15+19 = 57$ ways to choose a project.

3) In a version of the computer language "BASIC", the name of a variable is a string of one or two alpha numeric characters, where upper case and lower case letters are not distiguishing more over a variable name must begin with a letter and must be different from the five strings of two characters that are reserved for programming use. Howmany different variable names are there in this version of BASIC?

Sol. Let V equal to the no.of different variable names in this version of BASIC. Let V_1 be the no.of these that are one character long and V_2 be the no.of these that are two characters long.

Then by the sum rule $V = V_1 + V_2$. Note that $V_1 = 26$, because a one character variable name must be a letter.

Furthermore by the product rule there are $26 \cdot 36$ strings of length two that begin with a letter and ends with an alphanumeric character. However 5 of these are excluded,

So $V_2 = 26 \cdot 36 - 5$

$= 931$

Hence there are $V = V_1 + V_2$

$= 26 + 931 = 957$

different names for variables in this ~~of~~ version of BASIC.

4) Each user on a computer system has a password, which is 6 to 8 characters long, where each character is an upper case letter or a digit. Each password must contain atleast one digit. How many possible passwords are there?

Sol. Let P be the total no. of possible passwords, and let P_6, P_7, P_8 denote the no. of possible passwords of length 6, 7 and 8 respectively. By the sum rule $P = P_6 + P_7 + P_8$. Finding P_6 directly is difficult, it is easier to find the the no. of strings of uppercase letters and digits that are 6 characters long, including those with no digits and subtract from this the no. of strings with no digits.

By the product rule, the no. of strings of 6 characters is 36^6 and the no. of strings with no digits is 26^6.

Hence $P_6 = 36^6 - 26^6$

$= 1867866560$

Similarly $P_7 = 36^7 - 26^7$

$= 70332353920$

$P_8 = 36^8 - 26^8$

$= 2088270645 76$

$\therefore P = P_6 + P_7 + P_8$

$= 2684483063360$

* ## Inclusion & Exclusion Principle:-

Suppose that a task can be done in n_1 or in n_2 ways, but some of the set of n_1 ways to do the task are the same as some of the n_2 other ways to do the task. In this situation we cannot use the sum rule to count the no. of ways to do the task. Adding the no. of ways to do the task in these two ways leads to an over count. To correctly count the no. of ways to do the task, we add the no. of ways to do it in one way and the no. of ways to do it in other way., and then subtract the no. of ways to do the task in a way. i.e; both among the set of n_1 ways and the set of n_2 ways. This technique is called the principle of inclusion - exclusion.

i.e; $|A_1 \cup A_2| = |A_1| + |A_2| - |A_1 \cap A_2|$

eg:- How many bit strings of length 8 either start with a '1' bit or end with the two bits "00".

Sol. We can construct a bit string of length 8 i.e; either start with a 1 bit (or) ends with two bits 00. The first task in constructing a bit string of length 8 begining with a 1 bit can be done in $2^7 = 128$ ways.

The second task constructing a bit string of length 8 ending with two bits 00 can be done in $2^6 = 64$ ways.

Both tasks, constructing a bit string of length 8 the begins with 1 and ends with 00 can be done in $2^5 = 32$ ways.

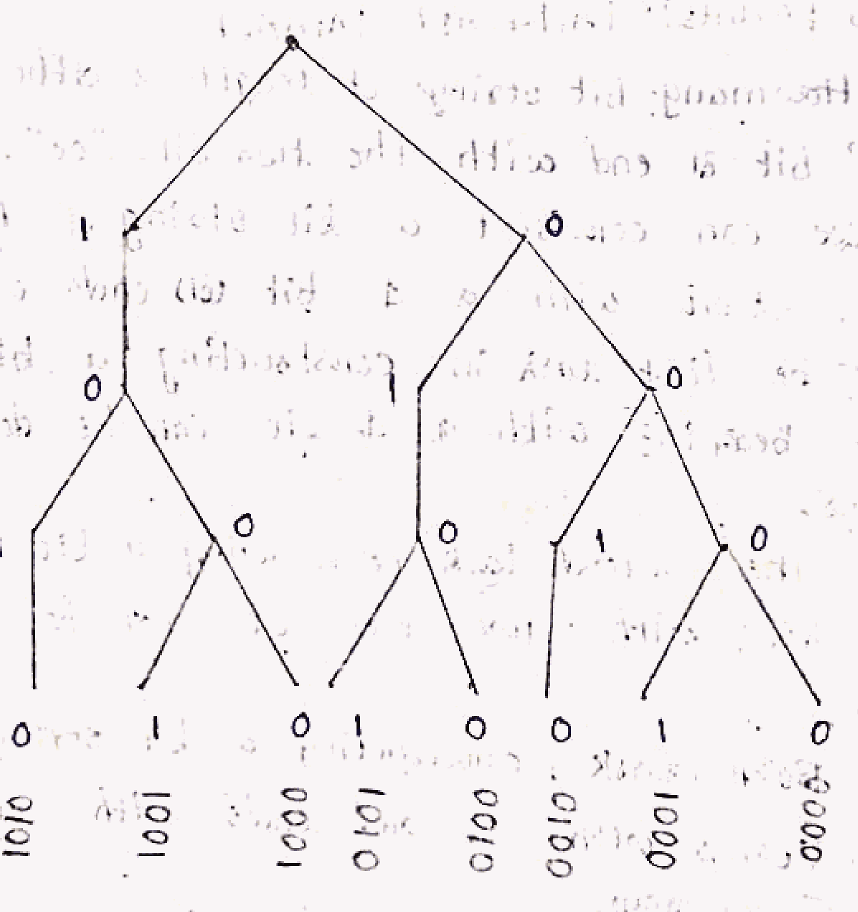
$\therefore$ By inclusion & exclusion principle the no. of bit strings of length 8 either start with a bit 1 or ends with two bits 00 are $128 + 64 - 32$
$= 160$

* Tree diagram :-

Counting problems can be solved using tree diagrams. A tree consists of a rule, a no. of branches leaving the rule, and possible additional branches leaving the end points of other branches. To use trees in counting, we use branch to represent each possible choice. we represent the possible outcomes by the leaves, which are the end points of branches not having other branches starting at them.

eg:- How many bit strings of length 4 donot have two consecutive 1s.

The below diagram displays all the bit strings of length 4 that are not containing consecutive 1s



* Pigeonhole – Principle :-

The pigeonhole principle statement that if there are more pigeons than pigeon holes, then there must be atleast one pigeon hole with atleast two pigeons in it.

Threorem:-

If k is a +ve integer and $(k+1)$ or more objects are placed into k boxes, then there is atleast one box containing two or more of the objects.

Proof:- we will prove this principle using a proof by contraposition. Suppose that non of the k boxes contains more than one object then the total no.of objects would be atmost k. This is a controduction becouse there are atleast $k+1$ objects.

$\therefore$ Atleast one box contain two or more of the objects.

eg:- Howmany students must be in a class to gaurantee that atleast two students receive the same score on the final exam, if the exam is graded on a scale from 0 to 100 points.

Sol. There are 101 possible scores on the final exam. The pigeonhole principle says that among any 102 students there must be atleast two students with the same score.

* The generalized pigeonhole principle:-

Theorem:-

If N objects are placed into k boxes, then there is atleast one box containing atleast $\lceil N/k \rceil$ objects.

Proof:- we will use a proof by controduction suppose that none of the boxes contains $\lceil N/k \rceil - 1$ then the total no.of objects is atmost $k(\lceil N/k \rceil - 1) < k((\lceil N/k \rceil + 1) - 1) = N$

This is a controduction because a total of N objects hence the theorem is true.

eg:- Among 100 people there are atleast $\lceil 100/12 \rceil = 9$ who were born in the same month.

Sol. Note:- when we have N objects, the generalized pigeonhole principle says that there must be atleast 'r' objects in one of the boxes as long as $\lceil N/K \rceil \geq r$. The smallest integer N with $N/K > r-1$ namely $N = K(r-1)+1$ is the smallest integer satisfying the inequality $\lceil N/K \rceil \geq r$.

1) What is the minimum no. of students required in a DMS class to be sure that atleast 6 will receive the grade. If there are 5 possible grades.

Sol. Here $K=5$, $r=6$

Since we know that $N = K(r-1)+1$
$= 5(6-1)+1$
$= 25+1$
$= 26$

$\therefore$ 26 is the minimum no. of students needed to show that atleast 6 students will receive the same grade.

2) How many cars must be selected from a standard deck of 52 cards to gaurantee that 3 cards of the same suit are choosen.

Sol. Here $K=4$ and $r=3$

Since $N = K(r-1)+1$
$N = 4(3-1)+1$
$N = 8+1$
$N = 9$

* **Permutations:-**

A permutation of a set of distint object is an ordered arrangement of objects. An ordered arrangement of 'r' elements of a set is called an "r-permutation"

eg:- Let $S=\{1,2,3\}$ the arrangement of $\{3,1,2\}$ is a permutation of S. The arrangement of $\{3,2\}$ is a two permutation of 5.

The no. of r-permutations of a set with n elements is denoted by $p(n,r)$ where $p(n,r) = \frac{n!}{(n-r)!}$.

Theorem:-

1) The no. of r permutations of a set with n distinct elements is $p(n,r) = n(n-1)(n-2)\cdots(n-r+1)$.

Proof:-

The first element of the permutation can be choosen in n ways. Since there n elements in the set.

There are $(n-1)$ ways to choose the second element of the permutation. similarly there are $(n-2)$ ways to choose the 3^{rd} element and so on. Finally there are exactly $n-(r-1) = n-r+1$ ways to choose the r^{th} element.

consequently by the product rule there are

$$n(n-1)(n-2)\cdots(n-r+1) = \frac{n!}{(n-r)!}.$$

In particular $p(n,n) = \frac{n!}{(n-n)!} = n!$

2) How many ways are there to select a first prize winner, second prize winner and 3^{rd} prize winner from 100 different people who have entered a contest?

Sol. Because it matters which person wins which price, the number of ways to pick the three prize winners is the no. of ordered selections of 3 elements from a set of 100 elements.

i.e; the no. of 3-permutations of a set of 100 elements

$\therefore$ The answer is $p(100,3) = n(n-1)(n-2)$

$= 100(100-1)(100-2)$

$= 970200$

3) Suppose that there are 8 runners in a race. The winner receives a gold medal, the second place finisher receives a silver medal and the 3rd place finisher receives a bronze medal. How many different ways are there to award these medals, if all possible outcomes of the race can occur and there are no ties.

Sol. The number of different ways to award the medals is the no. of 3 permutations of a set with 8 elements. Hence there are $P(8,3) = 8 \times 7 \times 6 = 336$ possible ways.

4) How many permutations of the letters ABCDEFGH contained the string ABC.

Sol:- Because the letters ABC must occur as a block, we can find the answer by finding the no. of permutations of 6 objects.

$\therefore$ There are $6! = 720$ permutations of the letters A B C D E F G H in which ABC occurs as a block.

* Combinations:-

An r-combination of elements of a set is an unordered selection of r elements from the set. thus an r combination is simply a subset of a set with r elements.

The number of r combinations of a set with n distinct elements is denoted by $C(n,r) = \dfrac{n!}{r!(n-r)!}$

eg:- we can see the $C(4,2) = 6$. since two-combinations of $\{a,b,c,d\}$ are $\{a,b\}, \{b,c\}, \{c,d\}, \{d,a\}, \{a,c\}, \{b,d\}$

Theorem:-

1) The number of r-combinations of a set with n elements, where n is a non-negative integer, r is an integer with

$0 \leq r \leq n$, equals $C(n,r) = \dfrac{n!}{r!\,(n-r)!}$

Proof:- The r-permutations of the set can be obtained by forming $C(n,r)$ a r-combinations of the set and then ordering the elements in each r-combination, which can be done in $P(r,r)$ ways. Consiquently $P(n,r) = C(n,r) \cdot P(r,r)$

This implies that $C(n,r) = \dfrac{P(n,r)}{P(r,r)}$

$$= \frac{\dfrac{n!}{(n-r)!}}{r!} = \frac{n!}{r!\,(n-r)!}$$

Theorem:-

2) Let n & r be non negative integers with $0 \leq r \leq n$, then $^nC_r = {}^nC_{n-r}$

Proof:- since we know that $^nC_r = \dfrac{n!}{r!\,(n-r)!}$

$$\text{Now } {}^nC_{n-r} = \frac{n!}{(n-r)!\,(n-(n-r))!}$$

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

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

$$= {}^nC_r$$

3):- How many ways are there to select five players from a 10 member tennis team to make a trip to a match at another school.

Sol. The answer is given the no. of 5-combinations of a set with 10 element that is $C(10,5) = \dfrac{10!}{5!\,5!} = 252$

4) A group of 30 people have been trained as astronauts to go on the first machine to Mars. How many ways are there to select a crew of 6 people to go on this machin

The answer is $C(30,6) = \frac{30!}{6!(30-6)!} = \frac{30!}{6!\cdot 24!}$

5) Suppose that there are 9 faculty member in the mathematics and 11 in the computer science department. How many ways are there to select a committee to develop a DMS course at a school, if the committee is to consists of 3 faculty members from the Mathematics department and 4 from computer science department?

Sol. By the product rule the answer is the product of the no. of 3-combinations of a set with 9 elements & the no. of 4-combinations of a set with 11 elements. Then the no. of ways to select the committee is

$C(9,3) \times C(11,4)$

$= \frac{9!}{3!\cdot 6!} \cdot \frac{11!}{4!\cdot 7!}$

* Permutations with repetations:-

eg:- How many strings of length r can be formed from the english alphabet.

Sol. By the product rule, because there are 26 letters and because each letter can be used repeatedly then there are $(26)^r$ strings of length r.

Theorem:-

The no. of r-permutations of a set of n objects with repetation allowed is n^r.

Sol. There are n ways to select an element of the set for each of the 'r' positions in the r-permutation when repetation is allowed, because for each choice all n objects are available. Hence by the product rule there are n^r r-permutations when repetation is allowed.

* **Combinations with repetitions:-**

Theorem:-

There are $c(n+r-1, r) = c(n+r-1, n-1)$ are r-combinations from a set with n elements when repetition of elements is allowed.

Proof:- Each r-combinations of a set with n elements when repetition is allowed can be represented by a list of $n-1$ bars and r stars. The $n-1$ bars are used to mark of n different cells, with the ith cell containing a star for each time the ith element of the set occurs in the combination. For instance a six-combination of a set with 4 elements is represented with 3 bars and 6 stars. Here $**|*||***$ represents the combination consisting exactly two of the first element, one of the second element, none of the third and 3 of the fourth element of the set.

Each different list containing $(n-1)$ bars and r stars corresponding to an r-combination of the set with n elements, when repetition is allowed. The no. of such lists are $c(n-1+r, r)$, because is each list corresponds to a choice of the r positions to place the r stars from $(n-1+r)$ positions that contain r stars and $(n-1)$ bars. The no. of such lists is also equal to $c(n-1+r, n-1)$

Since we know that $^nc_r = {}^nc_{n-r}$.

Eg 1) Suppose that a cookie shop as 4 different kinds of cookies. How many different ways 6 cookies be choosen? Assume that only the type of cookie, and not the individual cookies or the order in which they are choosen matters.

Sol. The no. of ways to choose 6 cookies is the no. of 6-combinations of a set with 4 elements. This is equals to $C(4+6-1,6) = C(9,6) = 84$

* **Permutations with indistinguishable objects :-**

Theorem :-

The no. of different permutations of n objects there are n_1 indistinguishable objects of type 1, n_2 indistinguishable objects of type 2 and so on and n_k indistinguishable objects of type k, is $\dfrac{n!}{n_1!\,n_2!\cdots n_k!}$

Proof :- To determine the no. of permutations, first note that the n_1 objects of type 1 can be placed among the n positions in $C(n,n_1)$ ways, leaving $(n-n_1)$ positions free then the object of type 2 can be placed in $C(n-n_1, n_2)$ ways, leaving $(n-n_1-n_2)$ positions free. then continue placing the objects of type 3 and so on. type $k-1$ until the last stage n_k objects of type k can be placed in $C(n-n_1-n_2\cdots-n_{k-1}, n_k)$ ways.

Hence by the product rule the total no. of different permutations is

$$C(n,n_1)\cdot C(n-n_1,n_2)\cdots\cdots C(n-n_1-n_2\cdots\cdots-n_{k-1},n_k)$$

$$= \frac{n!}{n_1!\,\cancel{(n-n_1)!}}\cdot\frac{\cancel{(n-n_1)!}}{n_2!\,\cancel{(n-n_1-n_2)!}}\cdot\frac{\cancel{(n-n_1-n_2)!}}{n_3!\,(n-n_1-n_2-n_3)!}\cdots\cdots$$

$$\cdot\frac{(n-n_1-\cdots-n_{k-1})!}{n_k!\,(n-n_1-\cdots\cdots-n_{k-1}-n_k)!}$$

$$= \frac{n!}{n_1!}\cdot\frac{1}{n_2!}\cdot\frac{1}{n_3!}\cdots\frac{1}{n_k!\,0!} = \frac{n!}{n_1!\,n_2!\cdots n_k!}$$

eg

1) How many different strings can be made by reordering the letters of the word "SUCCESS."

Sol An the given word there are $n=7$ letters totally. and 's' occurs 3 times i.e; $n_1=3$. Similarly "c" occurs 2 times i.e; $n_2=2$, U & E occurs 1 time i.e; $n_3=1$ & $n_4=1$ respectively.

∴ The no. of different strings can be made by using the letters in the given word $= \dfrac{n!}{n_1!n_2!n_3!n_4!}$

$$= \frac{7!}{3!2!1!1!}$$

$$= \frac{7\times6\times5\times\cancel{4}^{2}\times\cancel{3}\times\cancel{2}\times1}{\cancel{3\times2}\times\cancel{2}\times1}$$

$$= 420$$

I

* Recurrence Relations :-

A recurrence relation for the sequence $\{a_n\}$ is an equation that expresses an interms of one or more of the previous terms of the sequence, Namely $a_0, a_1, \dots a_{n-1}$ for all integers $n \geq n_0$ where n_0 is a non negative integer. A sequence is called a solution of recurrence relation if its terms satisfy the recurrence relation.

eg:- Let $\{a_n\}$ be a sequence that satisfies the recurrence relation $a_n = a_{n-1} - a_{n-2}$ for $n=2,3,\dots$ suppose $a_0=3$ and $a_1=5$ what are a_2 & a_3

Sol. Since $a_n = a_{n-1} - a_{n-2}$

and $a_0=3$, $a_1=5$

Now $a_2 = a_1 - a_0 = 5-3 = 2$

|||ly $a_3 = a_2 - a_1 = 2-5 = -3$

Initial condition:-

The initial condition for a sequence specify a term that precedes the first term where the recurrence relation takes effect. For example $a_0=3$ & $a_1=5$ are the initial conditions in the above example.

eg:-

1) Suppose that a person deposites 10000 dalors in a Savings account at a bank yielding 11% per year with interest compounded annually. How much will be in the account are after 30 years.

Sol To solve this problem P_n denoteds the amount in the account after n years. Because the amount in the account after n years equals to the amount in the account after (n-1) years + Interest for the nth year, we can see that the sequence $\{P_n\}$ Satisfies the recurrence relation.

$$P_n = P_{n-1} + (0.11) P_{n-1} = (1.11) P_{n-1}$$

The initial condition is $P_0 = 10{,}000$

we can use an iterative approach to find a formula for P_n

$$P_1 = (1.11) P_0$$

$$P_2 = (1.11) P_1 = (1.11)^2 P_0$$

$$\vdots$$

$$P_n = (1.11)^n P_0$$

∴ The amount in the a/c after 30 years $= P_{30}$

∴ $P_{30} = (1.11)^{30} \times (10{,}000)$

$= 228922.97$

* Solving recurrence relation:-

Linear Homogeneous Recurrence Relation:-

A linear homogeneous recurrence relation of degree k with constant coefficients is a recurrence relation of the form $a_n = c_1 a_{n-1} + c_2 a_{n-2} + \cdots + c_k a_{n-k}$. where $c_1, c_2, \ldots c_k$ are real numbers and $c_k \neq 0$.

example:-

1) The recurrence relation $P_n = (1.11)P_{n-1}$ is a linear homogeneous recurrence relation of degree 1.

2) $f_n = f_{n-1} + f_{n-2}$ is a linear homogeneous recurrence relation of degree 2.

3) The recurrence relation $a_n = a_{n-1} + a^2_{n-2}$ is not a linear

4) The recurrence relation $h_n = 2h_{n-1} + 1$ is not homogeneous

* characteristic equation:-

The basic approach for solving linear homogeneous recurrence relation is to look for the solutions of the form $a_n = r^n$ where r is a constant note that $a_n = r^n$ is a solution of the recurrence relation $a_n = c_1 a_{n-1} + c_2 a_{n-2} + \cdots + c_k a_{n-k}$ iff $r^n = c_1 r^{n-1} + c_2 r^{n-2} + \cdots + c_k r^{n-k}$. when both sides of the equation are divided by r^{n-k} and the right hand side is subtracted from the left, we obtain the equation $r^k - c_1 r^{k-1} - c_2 r^{k-2} - \cdots - c_{k-1} r - c_k = 0$

The above equation is called the characteristic equation of the recurrence relation.

Note:- Let c_1 & c_2 be real numbers, suppose that $r^2 - c_1 r - c_2 = 0$ has two distinct roots r_1 and r_2, then the sequence $\{a_n\}$ is a solution of the recurrence relation $a_n = c_1 a_{n-1} + c_2 a_{n-2}$ if and only if $a_n = \alpha_1 r_1^n + \alpha_2 r_2^n$ for $n = 0, 1, 2, \ldots$ where α_1, α_2 are constants.

example:-

1) what is the solution of recurrence relation $a_n = a_{n-1} + 2a_{n-2}$ with $a_0 = 2, a_1 = 7$

Sol. Since the given recurrence relation is $a_n = a_{n-1} + 2a_{n-2}$

$$\Rightarrow a_n - a_{n-1} - 2a_{n-2} = 0$$

The characteristic equation of the above equation is

$$r^2 - r - 2 = 0$$

$$\Rightarrow r^2 - 2r + r - 2 = 0$$

$$\Rightarrow r(r-2) + 1(r-2) = 0$$

$$\Rightarrow (r-2)(r+1) = 0$$

$$\Rightarrow r_1 = 2, r_2 = -1$$

$\therefore$ The solution for the above equation is $a_n = \alpha_1 r_1^n + \alpha_2 r_2^n$

i.e; $a_n = \alpha_1 2^n + \alpha_2 (-1)^n \longrightarrow (1)$

Since $a_0 = 2$

From (1) $\Rightarrow a_0 = \alpha_1 2^0 + \alpha_2 (-1)^0$

$$2 = \alpha_1 + \alpha_2 \longrightarrow (2)$$

$a_1 = 7$

from (1) $\Rightarrow a_1 = \alpha_1 2^1 + \alpha_2 (-1)^1$

$$7 = 2\alpha_1 - \alpha_2 \longrightarrow (3)$$

(2) + (3)

$$\begin{aligned} \alpha_1 + \cancel{\alpha_2} &= 2 \\ 2\alpha_1 - \cancel{\alpha_2} &= 7 \\ \hline 3\alpha_1 &= 9 \\ \alpha_1 &= 3 \end{aligned}$$

$\therefore$ from (2) $\alpha_1 + \alpha_2 = 2$

$$3 + \alpha_2 = 2$$

$$\alpha_2 = -1$$

$\therefore$ The required solution is $a_n = 3\,2^n - 1(-1)^n$

Note: 1) If $r_1 \neq r_2$ then $a_n = \alpha_1 r_1^n + \alpha_2 r_2^n$

2) If $r_1 = r_2$ then $a_n = (\alpha_1 + n\alpha_2) r_1^n$

1) what is the solution of the recurrence relation $a_n = 6a_{n-1} - 9a_{n-2}$ with initial conditions $a_0 = 1$ and $a_1 = 6$.

Sol. Since the given recurrence relation is $a_n = 6a_{n-1} - 9a_{n-2}$

The characteristic equation of above equation is

$$a_n - 6a_{n-1} + 9a_{n-2} = 0 \longrightarrow (1)$$

i.e; $r^2 - 6r + 9 = 0$

$\Rightarrow r^2 - 3r - 3r + 9 = 0$

$\Rightarrow r(r-3) - 3(r-3) = 0$

$\Rightarrow r = 3, 3$

$\therefore r_1 = 3, r_2 = 3$

$\therefore$ The solution for the above equation is $a_n = \cancel{\alpha_1 r_1^n + \alpha_2 r_2^n}$

i.e; $\cancel{a_n = \alpha_1 3^n + \alpha_2 3^n}$ $a_n = (\alpha_1 + n\alpha_2) r_1^n$ since $r_1 = r_2$

$$\Rightarrow a_n = (\alpha_1 + n\alpha_2).3^n \rightarrow ①$$

Since $a_0 = 1$

$\Rightarrow a_0 = (\alpha_1 + 0(\alpha_2))3^0$

$\Rightarrow a_0 = \alpha_1 . 3^0$

$\Rightarrow \alpha_1 . 3^0 = 1$

$\Rightarrow \alpha_1 = 1 \rightarrow ②$

Since $a_1 = 6$.

$\Rightarrow a_1 = (\alpha_1 + 1(\alpha_2))3^1$

$\Rightarrow 6 = (\alpha_1 + \alpha_2)3$

$\Rightarrow \alpha_1 + \alpha_2 = 2$

$\Rightarrow \alpha_2 = 2 - \alpha_1$

$\Rightarrow \alpha_2 = 2 - 1 = 1 \rightarrow ③$

Substitute ② & ③ in eq ①

$\therefore a_n = (1+n)3^n$

$\therefore$ The required solution is $a_n = (1+n)3^n$

2) Find an explicit solution for the febonacci series the sequence of febonacci numbers satisfies the recurrence relation $F_n = F_{n-1} + F_{n-2}$ with initial conditions $F_0 = 0$, $F_1 = 1$.

Sol. Since the given recurrence relation is $F_n = F_{n-1} + F_{n-2}$

$$\Rightarrow F_n - F_{n-1} - F_{n-2} = 0 \rightarrow ①$$

The characteristic equation of ① is $r^2 - r - 1 = 0$

$$r_1 = \frac{1 + \sqrt{1 - 4(1)(-1)}}{2(1)} = \frac{1+\sqrt{1+4}}{2} = \frac{1+\sqrt{5}}{2}$$

$$\gamma_2 = \frac{1-\sqrt{1+4}}{2} = \frac{1-\sqrt{5}}{2}$$

Now the solution for eq ① is $F_n = \alpha_1 \gamma_1^n + \alpha_2 \gamma_2^n$

$$F_n = \alpha_1\left(\frac{1+\sqrt{5}}{2}\right)^n + \alpha_2\left(\frac{1-\sqrt{5}}{2}\right)^n \rightarrow ②$$

since $F_0 = 0$

From ② $F_0 = \alpha_1\left(\frac{1+\sqrt{5}}{2}\right)^0 + \alpha_2\left(\frac{1-\sqrt{5}}{2}\right)^0$

$$\Rightarrow \alpha_1 + \alpha_2 = 0 \rightarrow ③$$

since $F_1 = 1$

From ② $F_1 = \alpha_1\left(\frac{1+\sqrt{5}}{2}\right)^1 + \alpha_2\left(\frac{1-\sqrt{5}}{2}\right)^1$

$$2 = \alpha_1(1+\sqrt{5}) + \alpha_2(1-\sqrt{5}) \rightarrow ④$$

from ③ $\alpha_2 = -\alpha_1$

substitute $\alpha_2 = -\alpha_1$ in eq ④

$\alpha_1(1+\sqrt{5}) - \alpha_1(1-\sqrt{5}) = 2$

$\alpha_1(1+\sqrt{5}-1+\sqrt{5}) = 2$

$\alpha_1(2\sqrt{5}) = 2$

$\alpha_1\sqrt{5} = 1$

$\alpha_1 = \frac{1}{\sqrt{5}}$

$\therefore \alpha_2 = -\alpha_1 = \frac{-1}{\sqrt{5}}$

$\therefore$ The required solution is $F_n = \frac{1}{\sqrt{5}}\left(\frac{1+\sqrt{5}}{2}\right)^n - \frac{1}{\sqrt{5}}\left(\frac{1-\sqrt{5}}{2}\right)^n$

3) Find the solution to the recurrence relation $a_n = 6a_{n-1} - 11a_{n-2} + 6a_{n-3}$ with initial conditions $a_0 = 2$, $a_1 = 5$, $a_2 = 15$.

Sol since $a_n = 6a_{n-1} - 11a_{n-2} + 6a_{n-3}$

$$\Rightarrow a_n - 6a_{n-1} + 11a_{n-2} - 6a_{n-3} = 0 \rightarrow ①$$

The characteristic equation of eq ① is

$$\gamma^3 - 6\gamma^2 + 11\gamma - 6 = 0$$

$$\begin{array}{c|cccc} 1 & 1 & -6 & 11 & -6 \\ & 0 & 1 & -5 & 6 \\ \hline & 1 & -5 & 6 & \boxed{0} \end{array}$$

$\Rightarrow (r-1)(r^2-5r+6)=0$

$\Rightarrow (r-1)(r^2-3r-2r+6)=0$

$\Rightarrow (r-1)(r(r-3)-2(r-3))=0$

$\Rightarrow (r-1)(r-2)(r-3)=0$

$\therefore r_1=1, r_2=2, r_3=3$

Now the solution for the equation ① is

$a_n=\alpha_1 r_1^n+\alpha_2 r_2^n+\alpha_3 r_3^n$

$a_n=\alpha_1(1)^n+\alpha_2(2)^n+\alpha_3(3)^n$

$a_n=\alpha_1+\alpha_2 \cdot 2^n+\alpha_3 \cdot 3^n \longrightarrow$ ②.

Since $a_0=2$

$\Rightarrow a_0=\alpha_1+\alpha_2 2^0+\alpha_3 3^0$

$\Rightarrow 2=\alpha_1+\alpha_2+\alpha_3 \longrightarrow$ ③

Since $a_1=5$

$\Rightarrow a_1=\alpha_1+\alpha_2 \cdot 2^1+\alpha_3 \cdot 3^1$

$\Rightarrow \alpha_1+2\alpha_2+3\alpha_3=5 \longrightarrow$ ④

Since $a_2=15$

$\Rightarrow a_2=\alpha_1+\alpha_2 \cdot 2^2+\alpha_3 \cdot 3^2$

$\Rightarrow \alpha_1+4\alpha_2+9\alpha_3=15 \longrightarrow$ ⑤

④ − ③

$$\begin{array}{r} \cancel{\alpha_1}+2\alpha_2+3\alpha_3=5 \\ -\cancel{\alpha_1} - \alpha_2 - \alpha_3 = -2 \\ \hline \alpha_2+2\alpha_3=3 \end{array} \longrightarrow ⑥$$

⑤ − ③

$$\begin{array}{r} \cancel{\alpha_1}+4\alpha_2+9\alpha_3=15 \\ -\cancel{\alpha_1} - \alpha_2 - \alpha_3 = -2 \\ \hline 3\alpha_2+8\alpha_3=13 \end{array} \longrightarrow ⑦$$

from ⑥ & ⑦

$$3\alpha_2 + 6\alpha_3 = 9$$
$$3\alpha_2 + 8\alpha_3 = 13$$
$$-\quad - \quad -$$

$$-2\alpha_3 = -4$$
$$\alpha_3 = 2$$

from ⑥ $\alpha_2 + 2\alpha_3 = 3$

$$\alpha_2 + 2(2) = 3$$
$$\alpha_2 + 4 = 3$$
$$\alpha_2 = -1$$

$\therefore$ The res from eq ③ $\alpha_1 + \alpha_2 + \alpha_3 = 2$

$$\alpha_1 - 1 + 2 = 2$$
$$\alpha_1 + 1 = 2$$
$$\alpha_1 = 1$$

$\therefore$ The required solution is $a_n = 1 + (-1)2^n + 2(3)^n$

$$a_n = 1 - 2^n + 2\cdot 3^n$$

* Linear non-homogeneous recurrence relation with constant coefficients:-

A linear non homogeneous recurrence relation with constant coefficient is of the form $a_n = c_1 a_{n-1} + c_2 a_{n-2} + \cdots + c_k a_{n-k} + F(n)$ where $c_1, c_2, \ldots c_k$ are real numbers and $F(n)$ is a function not identically '0' depending only on n. The recurrence relation $a_n = c_1 a_{n-1} + c_2 a_{n-2} + \cdots + c_k a_{n-k}$ is called the associated homogeneous recurrence relation.

eg:- The recurrence relations $a_n = a_{n-1} + 2^n$, $a_n = a_{n-1} + a_{n-2} + (n^2+n+1)$, $a_n = 3a_{n-1} + n.3^n$, $a_n = a_{n-1} + a_{n-2} + a_{n-3} + n!$ are linear non homogeneous recurrence relations with constant coefficients.

Theorem:-

If $\{a_n^{(p)}\}$ is a particular solution of the non homogeneous linear recurrence relation with constant coefficients.

$a_n = c_1 a_{n-1} + c_2 a_{n-2} + \cdots + c_k a_{n-k} + F(n)$ then every solution is of the form $\{a_n^{[p]} + a_n^{[h]}\}$, where $\{a_n^{[h]}\}$ is a solution of the associated homogeneous recurrence relation.

$a_n = c_1 a_{n-1} + c_2 a_{n-2} + \cdots + c_k a_{n-k}$

Problem:-

1) Find all solutions of the recurrence relation $a_n = 3a_{n-1} + 2n$ what is the solution with $a_1 = 3$.

Sol The required solution of recurrence relation is

$$a_n = a_n^{[h]} + a_n^{[p]} \longrightarrow ①$$

Here $a_n^{[h]}$ is the solution for the homogeneous part that is the associated homogeneous recurrence relation.

i.e; $a_n = 3a_{n-1}$

$$a_n - 3a_{n-1} = 0 \longrightarrow ②$$

The characteristic equation of eq ② is $r - 3 = 0$

$\Rightarrow r = 3$

$\therefore$ Now the solution for eq ② is $a_n^{[h]} = \alpha_1 3^n \rightarrow ③$

Next we have to find a particular polynomial solution.

i.e; $a_n^{[p]}$

Since $F(n) = 2n$ i.e; a polynomial in n of degree 1, the reasonable solution is a linear function in n say $P_n = cn + d$ where c, d are constants to demonstrate whether there are any solutions of this form. suppose that $P_n = cn + d$ then $a_n = 3a_{n-1} + 2n$ becomes

$$P_n = cn + d = 3a_{n-1} + 2n$$
$$= 3[c(n-1) + d] + 2n$$
$$= 3[cn - c + d] + 2n$$
$$= (3c + 2)n + 3(d - c)$$

Comparing the two terms

$c = 3c + 2$ and $3d - 3c = d$

$\Rightarrow 3c - c = -2$; $2d = 3c$

$\Rightarrow 2c = -2$; $2d = 3(-1)$

$\Rightarrow c = -1$; $d = -3/2$

$\therefore a_n^{[P]} = (-1)\cdot n - \frac{3}{2}$

$a_n^{[P]} = -n - \frac{3}{2} \longrightarrow ④$

Since from eq① $a_n = a_n^{[h]} + a_n^{[P]}$

$a_n = \alpha_1 \cdot 3^n - n - \frac{3}{2} \longrightarrow ⑤$

Since $a_1 = 3$

$\Rightarrow a_1 = \alpha_1 3^n - 1 - \frac{3}{2}$

$\Rightarrow 3 = \alpha_1 3^1 - 1 - \frac{3}{2}$

$\Rightarrow 6 = 6\alpha_1 - 2 - 3$

$\Rightarrow 6 = 6\alpha_1 - 5$

$\Rightarrow 6\alpha_1 = 11$

$\Rightarrow \alpha_1 = 11/6$

$\therefore$ The required solution is $a_n = \frac{11}{6} 3^n - n - \frac{3}{2}$

2) Find the solution of the recurrence relation $a_n = 5a_{n-1} - 6a_{n-2} + 7^n$

Sol. The required solution for the given recurrence relation is $a_n = a_n^{[h]} + a_n^{[P]} \longrightarrow ①$

Here $a_n^{[h]}$ is the solution of homogeneous part i.e; the associated homogeneous recurrence relation.

i.e; $a_n = 5a_{n-1} - 6a_{n-2}$

$\Rightarrow a_n - 5a_{n-1} + 6a_{n-2} = 0 \longrightarrow ②$

The characteristic equation of eq② is $r^2 - 5r + 6 = 0$

$r^2 - 3r - 2r + 6 = 0$

$r(r-3) - 2(r-3) = 0$

$r = 3, 2$

$r_1 = 2, r_2 = 3$

Now the solution of homogeneous part is $a_n^{[h]} = \alpha_1 2^n + \alpha_2 3^n \rightarrow (3)$

Next we have to find a particular polynomial solution

i.e; $a_n^{[P]}$

Since $F(n) = 7^n$.

The reasonable solution is $a_n^{[P]} = c.7^n$ where c is the constant.

Substituting the term of this sequence into the recurrence relation implies $c.7^n = 5(c.7^{n-1}) - 6(c.7^{n-2}) + 7^n$

$$= 5\left(c.\frac{7^n}{7}\right) - 6\left(c.\frac{7^n}{7^2}\right) + 7^n$$

$$= 7^n\left(\frac{5c}{7} - \frac{6c}{7^2} + 1\right)$$

Comparing the relative terms on both sides

$$c = \frac{5c}{7} - \frac{6c}{7^2} + 1$$

$$c = \frac{35c - 6c + 49}{49}$$

$$49c = 29c + 49$$

$$20c = 49$$

$$c = 49/20$$

$$\therefore a_n^{[P]} = \frac{49}{20}.7^n$$

$\therefore$ The required solution is $a_n = a_n^{[h]} + a_n^{[P]}$

$$a_n = \alpha_1 2^n + \alpha_2 3^n + \frac{49}{20}.7^n$$

* Divide and conquer algorithms and recurrence relations :-

Suppose that a recursive algorithm divides a problem of size n into 'a' subproblems, where each subproblem is of size 'n/b', also suppose that a total of g(n) extra operations are required in the conqure step of the algorithm to combine the solutions of the subproblems into solution of original problem. If f(n) represents the no. of operations required to solve the problem of size n, it follows the f satisfies the recurrence relation $f(n) = af(n/b) + g(n)$. This is called the divide and conquer recurrance relation.

eg:- Binary search $g.f(n) = f(n/2) + 2$.

Theorem:-

Let f be an increasing function that satisfies the recurrence relation $f(n) = af(n/b) + c$ whenever n is divisible by b, where $a \geq 1$, b is an integer > 1, c is a +ve real number then $f(n)$ is $\begin{cases} O(n^{\log_b a}) & \text{if } a > 1 \\ O(\log n) & \text{if } a = 1 \end{cases}$

further more when $n = b^k$, k is a +ve integer $f(n) = c_1 n^{\log_b a} + c_2$ where $c_1 = f(1) + c/(a-1)$ and $c_2 = -c/(a-1)$

Proof:- Suppose $n = b^k$ and $f(n) = af(n/b) + c$

$$\begin{aligned} \Rightarrow f(b^k) &= af(b^k/b) + c \\ &= af(b^{k-1}) + c \\ &= a[af(b^{k-2}) + c] + c \\ &= a^2 f(b^{k-2}) + ac + c \\ &\ \ \text{- - - - - - - - - -} \\ &\ \ \text{- - - - - - - - - -} \\ &= a^k f(b^{k-k}) + a^{k-1} \cdot c \\ &\quad + a^{k-2} \cdot c + \cdots + c \\ &= a^k f(1) + \\ &\quad c(a^{k-1} + a^{k-2} + \cdots + a + 1) \\ &= a^k f(1) + c \sum_{j=0}^{k-1} a^j \end{aligned}$$

First consider the case when $a = 1$

$$\begin{aligned} \Rightarrow f(b^k) &= 1^k \cdot f(1) + c \sum_{j=0}^{k-1} 1^j \\ &= f(1) + ck \end{aligned}$$

since $n = b^k$ apply "log" on both sides

$\log_b n = \log_b b^k$

$\log_b^n = K$

$\therefore f(n) = f(1) + c.\log_b^n$

$\Rightarrow f(n) \geq c.\log_b^n$

$\Rightarrow f(n)$ is $O(\log_b^n)$.

Suppose $a>1$

assume that $n=b^K$ then $f(n) = a^K.f(1) + c\sum_{j=0}^{K-1} a^j$

[The series is in G.P $a\left(\frac{r^n-1}{r-1}\right)$]

$$= a^K.f(1) + c\left(\frac{a^K-1}{a-1}\right)$$

$$= a^K\left[f(1) + \frac{c}{a-1}\right] - \frac{c}{a-1}$$

$$= c_1 a^K + c_2$$

since $n=b^K$ apply log on both sides

$$\log_b^n = \log_b b^K$$

$$\log_b^n = K$$

$$f(n) = c_1 a^{\log_b^n} + c_2$$

$$f(n) = c_1 n^{\log_b^a} + c_2$$

eg:- 1) Let $f(n) = 5.f(n/2)+3$ and $f(1) = 7$ estimate $f(n)$ if f is an increasing function.

Sol. It is in the form of $f(n) = a.f(n/b)+c$

Here $a=5$, $b=2$ and $c=3$

$\therefore f(n)$ is order of $O(n^{\log_2 5})$

2) Let $f(n) = 2.f(n/2)+2$

Here $a=2$, $b=2$, $c=2$

$\therefore f(n)$ is $O(n^{\log_2^2}) = O(n)$

* Master Theorem:-

Let f be an increasing function that satisfying the recurrence relation $f(n) = af(n/b) + c.n^d$ where $n=b^K$, K is a +ve Integer, $a \geq 1$, b is a +ve integer >1 and c, d are real numbers then $f(n)$ is ~~Od~~

$$\begin{cases} O(n^d) & \text{if } a<b^d \\ O(n^d \log n) & \text{if } a=b^d \\ O(n^{\log_b a}) & \text{if } a>b^d \end{cases}$$

eg:- 1) $f(n) = 2f(n/2) + n$

Sol. The given recurrence relation is in the form of $f(n) = af(n/b) + c.n^d$

Here $a=2, b=2, c=1, d=1$

Since $a = b^d$, $f(n)$ is $O(n^1 \log n) = O(n \log n)$

* **Generating Functions:-**

The generating function for the sequence $a_0, a_1, \cdots a_k \cdots$ of real numbers is the infinite series $G(x) = a_0 + a_1x + \cdots + a_kx^k + \cdots = \sum_{k=0}^{\infty} a_kx^k$.

eg:- what is the generating function for the sequence 1, 1, 1, 1, 1, 1.

Sol The generating function for the sequence is

$G(x) = 1 + x + x^2 + x^3 + x^4 + x^5 = 1\left(\frac{x^6-1}{x-1}\right) = \frac{x^6-1}{x-1}$ $\quad a\left(\frac{x^n-1}{x-1}\right)$

eg:- Let m be the +ve integer, let $a_k = C(m,k)$ for $k = 0, 1, 2, \ldots m$. what is the generating function for the sequence $a_0, a_1, \cdots a_m$

Sol. The generating function for this sequence is

$G(x) = a_0 + a_1x + a_2x^2 + \cdots + a_mx^m$

$= C(m,0) + C(m,1)x + C(m,2)x^2 + \cdots + C(m,m)x^m$

$= (1+x)^m$ $\quad \left[\begin{array}{l} (x+y)^n = {}^nC_0 x^n + {}^nC_1 x^{n-1}y + {}^nC_2 x^{n-2}y^2 \ldots \\ (1+x)^n = {}^nC_0 + {}^nC_1 x + {}^nC_2 x^2 + \ldots \end{array}\right]$

Note:- 1) $f(x) = \frac{1}{1-x}$ is the generating function of the sequence 1, 1, 1, ... since $\frac{1}{1-x} = 1 + x + x^2 + \cdots$

2) The function $f(x) = \frac{1}{1-ax}$ is the generating function

of the sequence $1, a, a^2, a^3, \ldots$ because $\frac{1}{1-ax} = 1+ax+a^2x^2 + a^3x^3 + \ldots$

* **Extended binomial coefficients:-**

Let u be a real number and k is a non -ve integer then the extended binomial coefficient $\binom{u}{k} =$
$$\begin{cases} u(u-1)\cdots(u-k+1)/k! & \text{if } k>0 \\ 1 & \text{if } k=0 \end{cases}$$

eg:- 1) Find the values of extended binomial coefficients $\binom{-2}{3}$ and $\binom{1/2}{3}$ i, Taking $u=-2, k=3$ and $k>0$

Sol. i, By the definition of extended binomial coefficients

$$\binom{-2}{3} = \frac{-2(-2-1)(-2-2)}{3!} = \frac{-\cancel{2}(-\cancel{3})(-4)}{\cancel{6}} = -4$$

ii, $\binom{1/2}{3}$

Taking $u=1/2$ and $k=3$, $k>0$

$$\binom{1/2}{3} = \frac{\frac{1}{2}\left(\frac{1}{2}-1\right)\left(\frac{1}{2}-2\right)}{3!} = \frac{\frac{1}{2}\left(\frac{-1}{2}\right)\left(\frac{-3}{2}\right)}{6} = \frac{\cancel{3}}{8\times\cancel{6}_2} = \frac{1}{16}$$

Note:- $\binom{-n}{r} = \frac{-n(-n-1)\cdots(-n-r+1)}{r!}$ (by the definition of extended binomial coefficient).

$$= \frac{(-1)^r \cdot n(n+1)\cdots(n+r-1)}{r!}$$

$$= \frac{(-1)^r \cdot (n+r-1)(n+r-2)\cdots(n+1)(n)}{r!}$$ by the commutative law of multiplication

Multiply both numerator and denominator with $(n-1)!$

$$= \frac{(-1)^r (n+r-1)(n+r-2)\cdots(n+1)(n)(n-1)!}{r! \cdot (n-1)!}$$

$$= \frac{(-1)^r \cdot (n+r-1)!}{r!\,(n-1)!} = (-1)^r \cdot C(n+r-1, r)$$

It is the alternative notation for binomial coefficient.

1) Find the generating function for $(1+x)^{-n}$ and $(1-x)^{-n}$ where n is a +ve integer, using extended binomial theorem.

Sol. Since we know that $(1+x)^n = \sum_{k=0}^{n} \binom{n}{k} \cdot x^k$

To find the generating function for $(1+x)^{-n}$, simply substitute '$-n$' in place of 'n'.

$$\Rightarrow (1+x)^{-n} = \sum_{k=0}^{n} \binom{-n}{k} x^k$$

$$= \sum_{k=0}^{n} (-1)^k \cdot c(n+k-1, k) \cdot x^k$$

Inorder to find the generating function for $(1-x)^{-n}$. Substitute '$-x$' in place of 'x', in the above equation

$$\Rightarrow (1-x)^{-n} = \sum_{k=0}^{n} (-1)^k c(n+k-1, k) \cdot (-x)^k$$

$$= \sum_{k=0}^{n} x^k c(n+k-1, k)$$

2) Find the no. of sequences of $e_1+e_2+e_3=17$ where e_1, e_2, e_3 are non negative integers with $2 \le e_1 \le 5$, $3 \le e_2 \le 6$, $4 \le e_3 \le 7$.

2 3 4 5 | 3 4 5 6 | 4 5 6 7

4 5 5 | 6 5 6 | 7 7

Sol. The no. of solutions with the indicated constraint is the coefficient of x^{17} in the expansion of $(x^2+x^3+x^4+x^5) \cdot (x^3+x^4+x^5+x^6)(x^4+x^5+x^6+x^7)$

$\therefore$ The ~~doe~~ quotient of x^{17} in this product is 3.

Hence there are 3 sequences.

3) In how many different ways can 8 identical cookies be distributed among 3 distinct children if each children receives atleast two cookies and no more than 4 cookies.

Sol. Since each children receives atleast two but no more than 4 cookies. For each children there is a factor $=(x^2+x^3+x^4)$

Since there are 3 children then the generating function is $(x^1+x^3+x^4)^3$. we need the coefficient of x^8 in this product. The coefficient = 6. There are 6 ways to distribute 8 cookies.

* Using generating functions to solve recurrence relations:-

1) Solve the recurrence relation $a_k = 3a_{k-1}$ for $k=1,2,3,\dots$ and initial condition $a_0 = 2$

Sol. Let $G(x)$ be the generating function for the sequence $\{a_k\}$, that is $G(x) = \sum_{k=0}^{\infty} a_k x^k$

First note that $x.G(x) = \sum_{k=0}^{\infty} a_k x^k . x = \sum_{k=0}^{\infty} a_k x^{k+1}$

$$= \sum_{k=1}^{\infty} a_{k-1} x^k$$

Now $G(x) - 3xG(x) = \sum_{k=0}^{\infty} a_k x^k - 3\sum_{k=1}^{\infty} a_{k-1} x^k$

$$= a_0 + \sum_{k=1}^{\infty} (a_k - 3a_{k-1}) x^k$$

$$= a_0 \qquad [\because a_k - 3a_{k-1} = 0$$

$$= 2$$

$$G(x)(1-3x) = 2$$

$$G(x) = \frac{2}{1-3x}$$

$$= 2.\sum_{k=0}^{\infty} (3x)^k \qquad \boxed{\because \frac{1}{1-ax} = 1+ax+a^2x^2+\dots}$$

$$= 2\sum_{k=0}^{\infty} 3^k.x^k$$

$$\therefore a_k = 2.3^k$$

2) Solve the recurrence relation $a_n = 8a_{n-1} + 10^{n-1}$ and initial condition $a_1 = 9$

Sol. Since $a_n = 8a_{n-1} + 10^{n-1}$

Multiply with x^n on bothsides

Then $a_n.x^n = 8a_{n-1}.x^n + 10^{n-1}.x^n \longrightarrow ①$

Let $G(x) = \sum_{n=0}^{\infty} a_n x^n$

Since $a_1 = 9$

$a_1 = 8a_{1-1} + 10^{1-1}$

$9 = 8a_0 + 1$

$8a_0 = 8$

$\therefore a_0 = 1$

Taking $\sum_{n=0}^{\infty}$ on bothsides in eq ①, we get

$$\sum_{n=0}^{\infty} a_n x^n = \sum_{n=0}^{\infty} 8a_{n-1} \cdot x^n + \sum_{n=0}^{\infty} 10^{n-1} \cdot x^n$$

$$\Rightarrow G(x) = 8\sum_{n=0}^{\infty} a_{n-1} \cdot x^n + \frac{1}{10}\sum_{n=0}^{\infty} 10^n \cdot x^n$$

$$\Rightarrow G(x) = 8x\sum_{n=1}^{\infty} a_{n-1} \cdot x^{n-1} + \frac{1}{10}\sum_{n=0}^{\infty} (10x)^n$$

$$\Rightarrow G(x) = 8x\sum_{n=1}^{\infty} a_{n-1} x^{n-1} + \frac{1}{10} \cdot \frac{1}{1-10x}$$

$$\frac{1}{1-ax} = 1+ax+a^2x^2 \ldots$$

$$\Rightarrow G(x) = 8x \cdot G(x) + \frac{1}{10} \cdot \frac{1}{(1-10x)}$$

$$\Rightarrow G(x)(1-8x) = \frac{1}{10(1-10x)}$$

$$\Rightarrow G(x) = \frac{1}{10}\left[\frac{1}{(1-10x)(1-8x)}\right]$$

$$\Rightarrow G(x) =$$

$$\Rightarrow \sum_{n=0}^{\infty} a_n x^n - a_0 = 8x\sum_{n=1}^{\infty} a_{n-1} x^{n-1} + x\sum_{n=1}^{\infty} 10^{n-1} \cdot x^{n-1}$$

$$\Rightarrow G(x) - 1 = 8x\sum_{n=0}^{\infty} a_n x^n + x\sum_{n=0}^{\infty} 10^n \cdot x^n$$

$$\Rightarrow G(x) - 1 = 8x \cdot G(x) + x \cdot \frac{1}{1-10x}$$

$$\Rightarrow G(x)[1-8x] = 1 + \frac{x}{1-10x}$$

$$\Rightarrow G(x)[1-8x] = \frac{1-10x+x}{1-10x}$$

$$\Rightarrow G(x)[1-8x] = \frac{1-9x}{1-10x}$$

$$\Rightarrow G(x) = \frac{1-9x}{(1-8x)(1-10x)}$$

$$= \frac{1}{2}\left[\frac{1}{1-8x} + \frac{1}{1-10x}\right]$$

[By using Partial fractions]

$$= \frac{1}{2}\left[\sum_{n=0}^{\infty} 8^n x^n + \sum_{n=0}^{\infty} 10^n \cdot x^n\right]$$

$$G(x) = \sum_{n=0}^{\infty} \frac{1}{2}(8^n + 10^n).x^n$$

$$a_n = \frac{1}{2}(8^n + 10^n)$$

3) Solve the recurrence relation $a_k = 7a_{k-1}$ with $a_0 = 5$

Sol: Let $G(x) = \sum_{k=0}^{\infty} a_k x^k$ be the generating function. over the sequence $a_0, a_1, \ldots$

Since $a_k = 7a_{k-1}$

Multiply with x^k on both sides

$\Rightarrow a_k \cdot x^k = 7a_{k-1} \cdot x^k$

Summing on bothsides starting with $k=1$

$$\sum_{k=1}^{\infty} a_k x^k = 7\sum_{k=1}^{\infty} a_{k-1} x^k$$

$$\Rightarrow G(x) - a_0 = 7x \sum_{k=1}^{\infty} a_{k-1} x^{k-1}$$

$$\Rightarrow G(x) - 5 = 7x\, G(x)$$

$$\Rightarrow G(x)(-7x+1) = 5$$

$$\Rightarrow G(x) = \frac{5}{1-7x}$$

$$\Rightarrow G(x) = 5 \cdot \sum_{k=0}^{\infty} 7^k . x^k$$

$$\Rightarrow G(x) = \sum_{k=0}^{\infty} (5.7^k).x^k$$

$$\therefore a_k = 5.7^k$$

4) Solve the recurrence relation $a_k = 3a_{k-1} + 2$ with $a_0 = 1$

Sol: Let $G(x) = \sum_{k=0}^{\infty} a_k x^k$ be the generating function over the sequence $a_0, a_1, \ldots$

Since $a_k = 3a_{k-1} + 2$

Multiply with x^k on bothsides

$$a_k \cdot x^k = 3a_{k-1} \cdot x^k + 2x^k$$

summing on bothsides starting with $k=1$

$$\sum_{k=1}^{\infty} a_k \cdot x^k = 3\sum_{k=1}^{\infty} a_{k-1} x^k + 2\sum_{k=1}^{\infty} x^k$$

$$\sum_{k=1}^{\infty} a_k \cdot x^k = 3x \sum_{k=1}^{\infty} a_{k-1} x^{k-1} + 2x \sum_{k=1}^{\infty} x^{k-1}$$

$$\Rightarrow G(x) - a_0 = 3x \cdot G(x) + 2x \sum_{k=0}^{\infty} x^k$$

$$\Rightarrow G(x) - 1 = 3x G(x) + 2x \cdot \frac{1}{1-x} \qquad \left[\sum_{k=0}^{k} x^k = \frac{1}{1-x}\right]$$

$$\Rightarrow G(x)(1-3x) = 1 + \frac{2x}{1-x}$$

$$\Rightarrow G(x)(1-3x) = \frac{1-x+2x}{1-x}$$

$$\Rightarrow G(x)(1-3x) = \frac{1+x}{1-x}$$

$$\Rightarrow G(x) = \frac{1+x}{(1-x)(1-3x)}$$

$$\Rightarrow G(x) = \left[\frac{2}{1-3x} - \frac{1}{1-x}\right]$$

$$= 2 \cdot \sum_{k=0}^{\infty} 3^k x^k - \sum_{k=0}^{\infty} x^k$$

$$= \sum_{k=0}^{\infty} (2 \cdot 3^k - 1) \cdot x^k$$

$$\therefore a_k = 2 \cdot 3^k - 1$$

$$\frac{A}{1-x} + \frac{B}{1-3x}$$

$$\frac{A(1-3x) + B(1-x)}{(1-x)(1-3x)}$$

$$A - 3Ax + B - Bx$$

$$(A+B) - (3A+B)x$$

$$A + B = 1$$

$$3A + B = -1$$

$$-2A = 2$$

$$A = -1$$

$$A + B = 1$$

$$B = 2$$

5) Solve the recurrence relation, $a_k = 3a_{k-1} + 4^{k-1}$ with $a_0 = 6$

Sol. Let $G(x) = \sum_{k=0}^{\infty} a_k x^k$ be the generating function over the sequence $a_0, a_1, \ldots$

Since $a_k = 3a_{k-1} + 4^{k-1}$

Multiplying with x^k on both sides

$$a_k x^k = 3a_{k-1} x^k + 4^{k-1} x^k$$

Summing on both sides with $k=1$

$$\sum_{k=1}^{\infty} a_k x^k = 3 \sum_{k=1}^{\infty} a_{k-1} x^k + \sum_{k=1}^{\infty} 4^{k-1} x^k$$

$$G(x) - a_0 = 3x \sum_{k=1}^{\infty} a_{k-1} x^{k-1} + x \sum_{k=1}^{\infty} 4^{k-1} x^{k-1}$$

$$G(x) - 6 = 3x \cdot G(x) + x \sum_{k=0}^{\infty} 4^k x^k$$

$$G(x)(1-3x) = 6 + x \frac{1}{1-4x}$$

$$G(x)(1-3x) = \frac{6(1-4x)+x}{1-4x}$$

$$\frac{A}{1-3x} + \frac{B}{1-4x}$$

$$G(x)(1-3x) = \frac{6-24x+x}{1-4x}$$

$$A(1-4x) + B(1-3x)$$

$$A+B - 4Ax - 3Bx$$

$$A+B - (4A+3B)x$$

$$G(x)(1-3x) = \frac{6-23x}{1-4x}$$

$$A+B = 6$$

$$4A+3B = 23$$

$$A+1 = 6$$

$$A = 5$$

$$4A+4B = 24$$

$$-B = -1$$

$$B = 1$$

$$G(x) = \frac{6-23x}{(1-3x)(1-4x)}$$

$$G(x) = \frac{5}{(1-3x)} + \frac{1}{(1-4x)}$$

$$= 5 \cdot \sum_{k=0}^{\infty} 3^k x^k + \sum_{k=0}^{\infty} 4^k x^k$$

$$= \sum_{k=0}^{\infty} (5 \cdot 3^k + 4^k) \cdot x^k$$

$$\therefore a_k = 5 \cdot 3^k + 4^k$$

6) Solve the recurrence relation $a_k = 5a_{k-1} - 6a_{k-2}$ with initial conditions $a_0 = 6$, $a_1 = 30$

Sol. Let $G(x) = \sum_{k=0}^{\infty} a_k x^k$ be the generating function over the sequence $a_0, a_1, \cdots$

Since $a_k = 5a_{k-1} - 6a_{k-2}$

Multiply on both sides with, x^k

$$a_k x^k = 5a_{k-1} x^k - 6a_{k-2} x^k$$

Summing on both sides starting with $k=2$

$$\sum_{k=2}^{\infty} a_k x^k = \sum_{k=2}^{\infty} 5a_{k-1} x^k - 6\sum_{k=2}^{\infty} a_{k-2} x^k$$

$$\Rightarrow \sum_{k=0}^{\infty} a_k x^k - a_0 - a_1 x = 5x \sum_{k=2}^{\infty} a_{k-1} x^{k-1} - 6x^2 \sum_{k=2}^{\infty} a_{k-2} x^{k-2}$$

$$\Rightarrow G(x) - 6 - 30x = 5x \left[\sum_{k=1}^{\infty} a_{k-1} x^{k-1} - a_0\right] - 6x^2 \sum_{k=0}^{\infty} a_k x^k$$

$\Rightarrow G(x) - 6 - 30x = 5x[G(x) - 6] - 6x^2 G(x)$

$\Rightarrow G(x) - 6 - 3\cancel{0}x = 5xG(x) - 3\cancel{0}x - 6x^2 G(x)$

$\Rightarrow G(x) - 5xG(x) + 6x^2 G(x) = 6$

$\Rightarrow G(x)(1 - 5x + 6x^2) = 6$

$\Rightarrow G(x) = \dfrac{6}{6x^2 - 5x + 1}$

$\Rightarrow G(x) = \dfrac{6}{6x^2 - 3x - 2x + 1}$

$\Rightarrow G(x) = \dfrac{6}{(3x-1)(2x-1)}$

$\Rightarrow G(x) = \dfrac{-18}{(3x-1)} + \dfrac{12}{(2x-1)}$

$\Rightarrow G(x) = +18 \sum_{k=0}^{\infty} 3^k x^k - 12 \sum_{k=0}^{\infty} 2^k x^k$

$= \sum_{k=0}^{\infty} (18\,3^k - 12\,2^k) x^k$

$\therefore a_k = 18 \cdot 3^k - 12 \cdot 2^k$

$\dfrac{A}{3x-1} + \dfrac{B}{2x-1}$

$A(2x-1) + B(3x-1)$

$2Ax - A + 3Bx - B$

$-(A+B) + (2A+3B)x$

$A+B = -6,\ 2A+3B = 0$

$2A+2B = -12$

$\boxed{B = 12}$

$A + B = -6$

$\boxed{A = -18}$

The Algebraic Structures And Morphisms

* Algebraic System:-

A system consisting of a set and, one or more n-ary operations on the set will be called an algebraic system. (or) simply an algebra will denote an algebraic system by $<S, f_1, f_2, \ldots>$ where S is a non-empty set and $f_1, f_2, f_3, \ldots$ are operations on S. Since the operations and relations on the set S define a structure on the elements of S, an algebraic system is called an algebraic structure.

Two algebraic structures $<X, o>$ and $<Y, *>$ are said to be of the same type whenever the n-ary operations 'o' & '*' have the same value of n.

eg:- Let I be the set of integers. consider the algebraic system $<I, +, \times>$ where + and $\times$ are operations of addition and multiplication on I. A list of properties of these operations given below.

(A-1): For any $a, b \in I$

$(a+b)+c = a+(b+c)$ (Associativity)

(A-2): For any $a, b \in I$

$a+b = b+a$ (commutative)

(A-3): $\exists$ a distinguished element $0 \in I$ Such that for any $a \in I$

$a+0 = 0+a = a$ (Identity)

(A-4): For each $a \in I$ $\exists$ an element in I defined by $-a$ Such that

$a+(-a) = 0$ (Inverse)

(M-1): For any $a, b, c \in I$

$(a \times b) \times c = a \times (b \times c)$ (Associativity)

(M-2): for any $a, b \in I$

$a \times b = b \times a$ (Commutativity)

(M-3): There exists a distinguished element $1 \in I$ such that for any $a \in I$.

$a \times 1 = 1 \times a = a$ (Identity)

(D): For any $a, b, c \in I$

$a \times (b \times c) = (a \times b) \times c$ (Distributivity)

(C): For $a, b, c \in I$ and $a \neq 0$

$a \times b = a \times c \Rightarrow b = c$ (Cancellation property)

The algebraic system $\langle I, +, \times \rangle$ should have been expressed as $\langle I, +, \times, 0, 1 \rangle$. Inorder to emphasize the fact that 0 and 1 are distinguished elements of I.

eg:- 1) Let R be the set of real numbers and '+' & 'x' be the operations of addition and multiplication on R. The algebraic system $\langle R, +, \times \rangle$ satisfies all the properties given for the system $\langle I, +, \times \rangle$

2) In the algebraic system $\langle N, +, \times \rangle$ where N is the set of natural numbers and the operations + and x have their usual meanings, all properties listed for $\langle I, +, \times \rangle$ except (A-4) are satisfied.

3) Let S be a non empty set and P(S) be the its power set. For any sets $A, B \in P(S)$, define the operations + and x on P(S) as $A + B = (A - B) \cup (B - A)$, $A \times B = A \cap B$. Now the algebraic system $\langle P(S), +, \times \rangle$ satisfies all the properties listed except (C).

Here elements ϕ and S are the identity elements for + and x respectively.

* **Homomorphism :-**

Let $\langle X, \circ \rangle$ and $\langle Y, * \rangle$ be two algebraic systems of the same type in the sense that both '$\circ$' & '$*$' are binary operations, a mapping $g: X \to Y$ is called a homomorphism or simply morphism from $\langle X, \circ \rangle$ to $\langle Y, * \rangle$ if for any $x_1, x_2 \in X$, $g(x_1 \circ x_2) = g(x_1) * g(x_2)$, if such function g exists then $\langle Y, * \rangle$ is a homomorphic image of $\langle X, \circ \rangle$ we must note that $g(X) \subseteq Y$.

* **Epimorphism :-**

Let g be a homomorphism from $\langle X, \circ \rangle$ to $\langle Y, * \rangle$ if $g: X \to Y$ is onto then g is called "epimorphism".

* **Monomorphism :-**

Let g be a homomorphism from $\langle X, \circ \rangle$ to $\langle Y, * \rangle$. If $g: X \to Y$ is one-one then g is called "Monomorphism"

* **Isomorphism :-**

Let g be a homomorphism from $\langle X, \circ \rangle$ to $\langle Y, * \rangle$ if $g: X \to Y$ is one-to-one & onto then g is called an "Isomorphism." Then $\langle X, \circ \rangle$ and $\langle Y, * \rangle$ are isomorphic.

* **Endomorphism and Automorphism :-**

Let $\langle X, \circ \rangle$ and $\langle Y, * \rangle$ be two algebraic systems such that $Y \subseteq X$. A homomorphism from $\langle X, \circ \rangle$ to $\langle Y, * \rangle$ in such a case is called an "endomorphism".

If $Y = X$ then an isomorphism from $\langle X, \circ \rangle$ to $\langle Y, * \rangle$ is called an "automorphism."

* **Substitution property :-**

Let $\langle X, \circ \rangle$ be an algebraic system in which '$\circ$' is a binary operation on X. Let us assume that E is an equivalence relation on X. The equivalence relation E is said to have the substitution property with respect to the operation '$\circ$' iff for any $x_1, x_2 \in X$, if $(x_1 E x_1') \wedge (x_2 E x_2') \Rightarrow (x_1 \circ x_2) E (x_1' \circ x_2')$ where $x_1', x_2' \in X$

* Congruence relation:-

Let $\langle X, \circ \rangle$ be an algebraic system and E be an equivalence relation on X. The relation E is called a congruence relation on $\langle X, \circ \rangle$ if E satisfies the substitution property w.r.to the operation '$\circ$'.

* Sub algebra:-

Let $\langle X, \circ \rangle$ be an algebraic system and $Y \subseteq X$ which is closed under the operation '$\circ$' then $\langle Y, \circ \rangle$ is called a subalgebra of $\langle X, \circ \rangle$

eg:- Let N is a set of natural numbers, E is a set of even natural numbers, O be a set of odd natural numbers then $\langle N, + \rangle$ is an algebraic system and $E \subseteq N$ and $\langle E, + \rangle$ is closed w.r.to '+' then $\langle E, + \rangle$ is a subalgebra where as O is not closed under '+' hence $\langle O, + \rangle$ is not a subalgebra.

* Direct product :-

Let $\langle X, \circ \rangle$ and $\langle Y, * \rangle$ be two algebraic systems of the same type, the algebraic system $\langle X \times Y, \oplus \rangle$ is the direct product of the algebras $\langle X, \circ \rangle$ and $\langle Y, * \rangle$, Provided the operation $\oplus$ is defined for any $x_1, x_2 \in X$ and $y_1, y_2 \in Y$ as $\langle x_1, y_1 \rangle \oplus \langle x_2, y_2 \rangle = \langle x_1 \circ x_2, y_1 * y_2 \rangle$.

The algebraic systems $\langle X, \circ \rangle$ and $\langle Y, * \rangle$ are called the factors of the algebra $\langle X \times Y, \oplus \rangle$.

1) Let $S = \{a, b, c\}$ and let $*$ denote a binary operation on S given in the following table A. Let $P = \{1, 2, 3\}$ and $\oplus$ be a binary operation on P given in the following table B, show that $\langle S, * \rangle$ and $\langle P, \oplus \rangle$ are isomorphism.

(A)

*	a	b	c
a	a	b	c
b	b	b	c
c	c	b	c

⊕	1	2	3
1	1	2	1
2	1	2	2
3	1	2	3

(B)

sol consider a mapping $g: S \to P$ such that $g(a)=1, g(b)=2$ and $g(c)=3$, obviously g is 1-1 and onto;

To show that g is a homomorphism. we have to P.T

$g(a*b) = g(a) \oplus g(b)$.

L.H.S $= g(a*b)$ R.H.S $= g(a) \oplus g(b)$

$= g(b)$ $= 1 \oplus 2$ from tables

$= 2$ $= 2$ (A) & (B)

$\therefore$ L.H.S = R.H.S

$\therefore$ g is homomorphic

Hence g is isomorphic and $\langle S, * \rangle$ and $\langle P, \oplus \rangle$ are also isomorphic with each other.

2) Given the algebraic systems $\langle N, + \rangle$ and $\langle Z_4, +_4 \rangle$, where N is the set of natural numbers and $+$ is the operation of addition on N. Show that there exists a homomorphism from $\langle N, + \rangle$ to $\langle Z_4, +_4 \rangle$. Here $Z_4 = \{[0],[1],[2],[3]\}$. de.

Sol Here $Z_4 = \{[0],[1],[2],[3]\}$

Define $g: N \to Z_4$

$g(a) = [a \pmod 4]$ for any $a \in N$

for $a, b \in N$ let $g(a) = [i]$, $g(b) = [j]$

then $g(a+b) = [(a+b) \bmod 4] = [(i+j) \bmod 4]$

$= [i] +_4 [j]$

$= g(a) +_4 g(b)$

$\therefore$ g is homomorphic.

* **Quotient algebra:-** (Quotient structure)

Let $\langle X, o \rangle$ be an algebraic system in which 'o' is a binary operation on X, let E be a congruence relation on $\langle X, o \rangle$. Since E is an equivalence relation on X, it partitions X into equivalence classes. The set of equivalence classes is the quotient set X/E. The algebraic system $\langle X/E, * \rangle$ is called the quotient algebra.

* **Natural Homomorphism :-**

The homomorphism $g_E : X \rightarrow X/E$ such that $g_E(x) = [x]$ for any $x \in X$, obviously $g_E(x_1 o x_2) = [x_1 o x_2] = [x_1] * [x_2] = g_E(x_1) * g_E(x_2)$, then the homomorphism g_E is called the Natural homomorphism, associated with the congruence relation E.

* **Semigroup :-**

Let S be a non-empty set and 'o' be a binary operation on S. The algebraic system $\langle S, o \rangle$ is called a Semigroup if the operation 'o' is associative. Another words $\langle S, o \rangle$ is a Semigroup if for any $x, y, z \in S$, $(xoy)oz = xo(yoz)$.

* **Monoid :-**

A Semigroup $\langle M, o \rangle$ with an identity element w.r.t the operation 'o' is called a "Monoid." In otherwords an algebraic system $\langle M, o \rangle$ is called a monoid if for any $x, y, z \in M$, $(xoy)oz = xo(yoz)$ and there exists an element $e \in M$ such that for any $x \in M$, $eox = xoe = x$.

eg:- 1) Let X be a non-empty set and X^X be the set of all mappings from X to X. Let 'o' denote the operation of composition of these mappings, i.e; for $f, g \in X^X$, $(fog)(x) = f(g(x)) \ \forall \ x \in X$ is in X^X. The algebra $\langle X^X, o \rangle$ is a Monoid ~~the~~ because the operation of composition is associative and the identity mapping $f(x) = x \ \forall \ x \in X$ is the identity operation.

2) Let S be a non-empty set and $P(S)$ is a power set of S. The algebras $\langle P(S), \cup \rangle$ and $\langle P(S), \cap \rangle$ are monoids, with identities ϕ and S respectively.

3) If N be the set of natural numbers then $\langle N, + \rangle$ and $\langle N, \times \rangle$ are monoids with identities 0 & 1 respectively on the other hand if E denotes the set of +ve even numbers

then $\langle I, + \rangle$ and $\langle I, \times \rangle$ are Semigroups, but not monoids

* **Cyclic monoid:-**

A monoid $\langle M, *, e \rangle$ is said to be cyclic if there exists an element $a \in M$ such that every element of M can be written as some powers of 'a' i.e; as $a^n \ \exists\ n \in N$. In this case the element 'a' is called the "generator of cyclic monoid."

A cyclic monoid is commutative because for any $b, c \in M$, $b = a^m$ and $c = a^n$ for some $m, n \in N$ so that $b * c = a^m * a^n$ $= a^{m+n} = a^{n+m} = a^n * a^m = c * b$

* **Free Monoid:-**

A monoid $\langle M^*, \circ \rangle$ is called a free monoid if M^* is generated by the alphabet M.

i.e; for example let $V = \{a, b\}$ and consider the strings ababbb and bb. The concatenation of these strings produce the string ababbbbb. It is easy to say that the concatenation is associative.

$\therefore \langle V^+, \circ \rangle$ is a semigroup called a free semigroup generated by the alphabet V. If we include the empty string $\wedge$ (CCOP), then so that $\wedge$ is the identity w.r.to the operation of concatenation and $\langle V^*, \circ, \wedge \rangle$ is a monoid i.e; a free monoid.

* **Semigroup Homomorphism:-**

Let $\langle S, * \rangle$ and $\langle T, \Delta \rangle$ be any two Semigroups a mapping $g : S \to T$ such that for any two elements $a, b \in S$, $g(a * b) = g(a) \Delta g(b)$ is called "semigroup homomorphism"

Semigroup homomorphism is called a Semigroup monomorphism, epimorphism, or isomorphism depending on whether the mapping is one-to-one, onto or one-to-one onto respectively.

* **Monoid Homomorphism:-**

Let $\langle M, *, e_M \rangle$ and $\langle T, \Delta, e_T \rangle$ be any two monoids

A mapping $g: S \to T$ such that for any two elements $a, b \in M$, $g(a * b) = g(a) \Delta g(b)$ and $g(e_M) = e_T$ is called a monoid homomorphism.

eg:- 1) Let $\langle N, + \rangle$ be the semigroup of natural numbers and $\langle S, * \rangle$ be the semigroup on $S = \{e, 0, 1\}$ with the operation $*$ given in the following table. A mapping $g: N \to S$ given by $g(0) = 1$ and $g(j) = 0$ for $j \neq 0$ is semigroup homomorphism. Although both $\langle N, + \rangle$ and $\langle S, * \rangle$ are monoids with identities $0, e$ respectively, g is not a monoid homomorphism because $g(0) \neq e$.

*	e	0	1
e	e	0	1
0	0	0	0
1	1	0	1

* Theorem:-

1) Let $\langle S, * \rangle$, $\langle T, \Delta \rangle$ and $\langle V, \oplus \rangle$ be semigroups and $g: S \to T$ and $h: T \to V$ be semigroup homomorphisms. Then $(h \circ g): S \to V$ is a semigroup homomorphism from $\langle S, * \rangle$ to $\langle V, \oplus \rangle$.

Proof:- Let $a, b \in S$ then $(h \circ g)(a * b) = h(g(a * b))$

g is semigroup homomorphism $\leftarrow$ $= h[g(a) \Delta g(b)]$

h is semigroup homomorphism $\leftarrow$ $= h(g(a)) \oplus h(g(b))$

$= h \circ g(a) \oplus h \circ g(b)$

$\therefore$ $(h \circ g)$ is a semigroup homomorphism.

2) Let $\langle S, * \rangle$ be a given semigroup. There exist a homomorphism $g: S \to S^S$, where $\langle S^S, \circ \rangle$ is a semigroup of functions from S to S under the operation of (left) composition.

Proof:- For any element $a \in S$, let $g(a) = f_a$ where $f_a \in S^S$ and f_a is defined by $f_a(b) = a * b$ for any $b \in S$.

Now $g(a*b) = f_{a*b}$

where $f_{a*b}(c) = (a*b)*c$

$= a*(b*c)$

$= f_a \circ (f_b(c))$

$= (f_a \circ f_b)(c)$

$\therefore g(a*b) = f_{a*b} = f_a \circ f_b = g(a) \circ g(b)$

This step shows that $g: S \to S^S$ is a homomorphism.

3) Let X be a set containing n elements, let X^* denote the free semigroup generated by X, and let $\langle S, \oplus \rangle$ be any other semigroup generated by any n generators then there exist a homomorphism $g: X^* \to S$.

Proof:-

Let Y be the set of n generators of S. Let $g: X \to Y$ be a one-to-one mapping given by $g(x_i) = y_i$ for $i = 1, 2, \dots n$. Now for any string $\alpha = x_1 x_2 \dots x_m$ of X^*, we define $g(\alpha) = g(x_1) \oplus g(x_2) \oplus \dots \oplus g(x_m)$.

From this definition it follows that for a string $\alpha, \beta \in X^*$, $g(\alpha\beta) = g(\alpha) \oplus g(\beta)$

So, g is the required homomorphism.

4) Let $\langle S, * \rangle$ and $\langle T, \Delta \rangle$ be two semigroups and g be a semigroup homomorphism from $\langle S, * \rangle$ to $\langle T, \Delta \rangle$. Corresponding to the homomorphism g, there exist a congruence relation R on $\langle S, * \rangle$ defined by xRy iff $g(x) = g(y)$ for $x, y \in S$. *

Proof:-

It is easy to say that R is an equivalence relation on S. Let $x_1, x_2, x_1', x_2' \in S$ such that $x_1 R x_1'$, and $x_2 R x_2'$.

From $g(x_1 * x_2) = g(x_1) \Delta g(x)$

$$= g(x_1') \Delta g(x_2')$$

$$= g(x_1' * x_2')$$

It follows that R is a congruence relation on $\langle S, * \rangle$.

5) Let $\langle S, * \rangle$ be a semigroup and R be a congruence relation on $\langle S, * \rangle$. The quotient set S/R is a semigroup $\langle S/R, \oplus \rangle$ where the operation $\oplus$ corresponds through the operation $*$ on S. Also there exist a homomorphism from $\langle S, * \rangle$ to $\langle S/R, \oplus \rangle$ called the Natural homomorphism.

Proof:-

Sol. For any $a \in S$, let $[a]$ denote the equivalence class corresponding to the congruence relation R. For $a, b \in S$ define an operation $\oplus$ on S/R given by $[a] \oplus [b] = [a * b]$.

The associativity of the operation $*$ gaurantees the associativity of the operation $\oplus$ on S/R, so that $\langle S/R, \oplus \rangle$ is a semigroup.

Next we define a mapping $g : S \to S/R$ given by $g(a) = [a]$ for any $a \in S$. Now for $a, b \in S$, $g(a * b) = [a * b] = [a] \oplus [b]$

$$= g(a) \oplus g(b)$$

So that g is a homomorphism from $\langle S, * \rangle$ to $\langle S/R, \oplus \rangle$ hence g is a natural homomorphism.

* Sub semigroups and Sub monoids :-

Let $\langle S, * \rangle$ be a semigroup and $T \subseteq S$. If the set T is closed under the operation $*$, then $\langle T, * \rangle$ is said to be a subsemigroup of $\langle S, * \rangle$. Similarly let $\langle M, *, e \rangle$ be a monoid and $T \subseteq M$. If T is closed under the operation $*$ and $e \in T$ then $\langle T, *, e \rangle$ is said to be a submonoid of $\langle M, *, e \rangle$

eg:- For the semigroup $\langle N, \times \rangle$, let T be the set of multiples of a positive integer m, then $\langle T, \times \rangle$ is a sub-

Semigroup of $\langle N, \times \rangle$.

1) Theorem:-

For any commutative monoid $\langle M, * \rangle$, the set of idempotent elements of M forms a submonoid.

Proof:- Since the identity element $e \in M$ is idempotent, $e \in S$, where S is the set of idempotents of M. Let $a, b \in S$ so that $a*a = a$; $b*b = b$.

Now $(a*b)*(a*b) = (a*b)*(b*a)$ [$\because$ M is commutative]

$= a*(b*b)*a$ [Associative]

$= a*(b*a)$ [idempotent]

$= a*(a*b)$ (commutative)

$= (a*a)*b$

$= a*b$

Since $a*b$ is an idempotent element then $(a*b) \in S$ whenever $a, b \in S$

$\therefore$ S is closed under the operation $*$

$\therefore \langle S, * \rangle$ is a submonoid.

* Groups:-

A group $\langle G, * \rangle$ is an algebraic system in which the binary operation $*$ on G satisfies three conditions.

1: For all $(x, y, z) \in G$, $x*(y*z) = (x*y)*z$ (Associativity)

2: There exists an element $e \in G \rightarrow$ for any $x \in G$

$x*e = e*x = x$ (Identity)

3: For every $x \in G$ $\exists$ an element denoted by $x^{-1} \in G$ such that $x^{-1}*x = x*x^{-1} = e$ (Inverse)

* Permutation on a set:-

Any one-to-one mapping of a set S onto S is called a "permutation of S."

* Theorem:-

Every row or column in the composition table of a group $\langle G, * \rangle$ is a permutation of the elements of G.

Proof:- As a first step we shall show that no row or column in the composition table can have an element of

G more than once.

Let us assume, that the row corresponding to an element $a \in G$ has two entries which are both k, that is assume that $a * b_1 = k$, $a * b_2 = k$, $b_1, b_2, k \in G$ and $b_1 \neq b_2$

From the cancellation property we have $b_1 = b_2$, which is a contradiction. A similar result holds for any column.

As a next step of our proof, we S.T every element of g appears in each row and column of the table of composition, for this consider the row corresponding to the element $a \in G$, and let 'b' be any element of G. since $b = a * (a^{-1} * b)$, b must appear in the row corresponding to the element $a \in G$. The same argument apply to every column of the table as well.

From the above result we can say that no two rows (or) columns are identical, it follows that every row as well as column of the composition table is obtained by a permutation of the elements of G and each row as well column is a distinct permutation.

* **Order of a group :-**

The order of a group $\langle G, * \rangle$ denoted by $|G|$, is the no. of elements of G, when G is finite.

* **Abelian group:-**

A group $\langle G, * \rangle$ in which the operation $*$ is commutative is called an abelian group.

eg:- 1) The algebra $\langle I, + \rangle$ is an abelian group.

2) The set of rational numbers excluding '0' is an abelian group under multiplication.

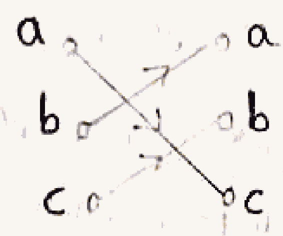
* **Permutation Groups:-**

Consider the set of all permutations of the elements of a finite set and define a binary operation on them. we shall consider those sets of permutations which form a group under this operation. such groups are called "Permutation groups."

Let $S = \{a, b, c\}$ be a set and let p denote a permutation of the elements of S. That is $p: S \to S$ is a bijective mapping. There are two convenient ways of discribing the permutation p. suppose that $p(a) = c$, $p(b) = a$, $p(c) = b$ then we may represent p as $p = \begin{pmatrix} a & b & c \\ c & a & b \end{pmatrix}$.

another method is to show p diagrametically.

i.e;



Let '$\Diamond$' denote a binary operation on S_n where the set S_n contains all the possible permutations of the elements of the given set S. For $i, j = 1, 2, \ldots$ we mean by $P_i \Diamond P_j$ is the permutation obtained by permuting the elements of S by an application of P_i followed by an application of the permutation P_j if we consider P_i & P_j as functions and let 'o' denote the left composition of functions then $P_i \Diamond P_j = P_j \circ P_i$ for $i, j = 1, 2, \ldots$ Accordingly $(P_i \Diamond P_j)(a) = (P_j \circ P_i)(a) = P_j(P_i(a))$.

The degree of a permutation group is the cardinality of the set on which the permutations are define.

eg:- $P_3 \Diamond P_5 = \begin{pmatrix} 1 & 2 & 3 \\ 3 & 2 & 1 \end{pmatrix} \Diamond \begin{pmatrix} 1 & 2 & 3 \\ 2 & 3 & 1 \end{pmatrix}$ $1 - 2 - 2$

$= \begin{pmatrix} 1 & 2 & 3 \\ 2 & 1 & 3 \end{pmatrix}$ $2 - 3 - 1$, $3 - 1 - 3$

In general the set S_n of all permutations of n elements is a permutation group $\langle S_n, \Diamond \rangle$ also called the Symmetric group.

* **Dihedral Group :-**

The set of all rigid rotations of a regular polygone of n sites under the composition $\Diamond$ is a group $<D_n, \Diamond>$. The group $<D_n, \Diamond>$ is a permutation group of degree n. The group D_n is called a "Dihedral group".

* **Cyclic group :-**

A group $<G, *>$ is said to cyclic if there exists an element $a \in G$ such that every element of G can be written as some power of 'a', i.e; a^n $\exists$ integer n. In such a case 'a' is called a generator of the group G.

A cyclic group is an abelian group.

1) **Theorem :-**

Let $<G, *>$ be a finite cyclic group generated by an element $a \in G$. If G is of order n i.e; $|G| = n$ then $a^n = e$ so that $G = \{a, a^2, a^3, \ldots a^n = e\}$ further more n is the least +ve integer for which $a^n = e$

Proof :- Let us assume that for some +ve integer $m < n$, $a^m = e$. Since G is a cyclic group, any element of G can be written as a^k $\exists$ $k \in I$. Now from Eclids algorithm we can write $k = mq + r$ where q is some integer and $0 \le r < m$. This means $a^k = a^{mq+r} = (a^m)^q * a^r$

$= e^q * a^r$

$= a^r$.

So that every element of G can be expressed as a^r for some $0 \le r < m$, thus implies that G has atmost 'm' distinct elements, i.e; $|G| = m < n$.

which is a controduction. Hence, $a^m = e$ for $m < n$ is not possible. As a next step we show that the elements $a, a^2, \ldots a^n$ are all distinct, where $a^n = e$. Assume to the contrary that $a^i = a^j$ for $i < j \le n$ this means the $a^{j-i} = e$ where $j - i < n$, which again is a controduction. Hence the theorem.

* **Sub group:-**

Let $<G,*>$ be a group and $S \subseteq G$ be such that it satisfies the following conditions.

1) $e \in S$, where e is the identity element of $<G,*>$.
2) for any $a \in S$, $a^{-1} \in S$.
3) for $a,b \in S$, $a*b \in S$

Note that $<S,*>$ itself is a group. For any group $<G,*>$ naturally $<\{e\},*\}$ and $<G,*>$ are "trivial subgroups" of $<G,*>$. All other subgroups of $<G,*>$ are called "proper subgroups."

1) **Theorem:-**

A subset $S \neq \phi$ of G is a subgroup of $<G,*>$ iff for any pair of elements $a,b \in S$, $a*b^{-1} \in S$

Proof:- Assume that S is a subgroup, it is clear that if $a,b \in S$ then a^{-1} and $b^{-1} \in S$.

Now $a*b^{-1} \in S$ (since by closure property).

To prove the converse let us assume that $a,b \in S$ and $a*b^{-1} \in S$ for any pair a,b

Taking $b=a$, $a*a^{-1} \in S$

$\Rightarrow e \in S$

From $e,a,b \in S$; we have $e*a^{-1} \in S$

$\Rightarrow a^{-1} \in S$

similarly $b^{-1} \in S$

Finally, because a and b^{-1} are in S we have $a*b \in S$

Hence $<S,*>$ is a subgroup of $<G,*>$.

* **Group Homomorphism:-**

Let $<G,*>$ and $<H,\Delta>$ be two groups, a mapping $g: G \rightarrow H$ is called a group homomorphism from $<G,*>$ to $<H,\Delta>$ if for any $a,b \in G$, $g(a*b) = g(a)\Delta g(b)$.

* Kernel :-

Let g be a group homomorphism from $\langle G,*\rangle$ to $\langle H,\Delta\rangle$. The set of elements of G which are mapped into e_H, the identity of H, is called the kernel of the homomorphism g, and denoted by $Ker(g)$

1) Theorem :-

The kernel of a homomorphism g from a group $\langle G,*\rangle$ to $\langle H,\Delta\rangle$ is a subgroup of $\langle G,*\rangle$.

Proof:- Since $g(e_G) = e_H$, $e_G \in ker(g)$. Also if $a,b \in ker(g)$

then $g(a) = g(b) = e_H$

then $g(a*b) = g(a) \Delta g(b)$

$= e_H \Delta e_H = e_H$

So that $(a*b) \in Ker(g)$. Finally if $a \in ker(g)$

then $g(a^{-1}) = [g(a)]^{-1} = [e_H]^{-1} = e_H$

Hence $a^{-1} \in ker(g)$

$\therefore$ $Ker(g)$ is a subgroup of $\langle G,*\rangle$

* Left coset:-

Let $\langle H,*\rangle$ be a subgroup of $\langle G,*\rangle$. For any $a \in G$, the set aH defined by $aH = \{a*h / h \in H\}$ is called the left coset of H in G determined by the element $a \in G$. The element a is called the representative element of the left coset aH

* Legrange's theorem:-

The order of a subgroup of a finite group divides the order of the group.

Proof:- The no. of left cosets of H in G is called the index of H in G. From Legrange's theorem we have the index K of G given by $K = |G|/|H|$. For any group $\langle G,*\rangle$ we know that the cyclic group generated by any elements of G must be a subgroup of G.

Let $a \in G$ and the cyclic group generated by 'a' by H, i.e; $H = \{a^i / i \in I\}$ if H is of order m, then according to Legrange's theorem m must divide n where n is the order of the group G. Hence from $a^m = e$ we also have $a^n = a^{mk} = (a^m)^k = e^k = e$.

Hence the theorem.

* Normal subgroup :-

A subgroup $\langle H, * \rangle$ of $\langle G, * \rangle$ is called a normal subgroup if for any $a \in G$, $aH = Ha$.

1) Theorem :-

Let $\langle G, * \rangle$ and $\langle H, \Delta \rangle$ be groups and $g: G \rightarrow H$ be a homomorphism then the ker(g) is a normal subgroup.

Proof :-

$K = \ker(g) = \{a / a \in G$ and $g(a) = e_H\}$ is a subgroup of $\langle G, * \rangle$.

Now for any $a \in G$ and $k \in K$, $g(a^{-1} * k * a) =$

$$g(a^{-1}) \Delta g(k) \Delta g(a)$$
$$= [g(a)]^{-1} \Delta e_H \Delta g(a)$$
$$= [g(a)]^{-1} \Delta g(a)$$
$$= e_H$$

Hence $(a^{-1} * k * a) \in K$, which shows the K is a normal subgroup.

* Direct Product :-

Let $\langle G, * \rangle$ and $\langle H, \Delta \rangle$ be two groups, the direct product of these two groups is the algebraic structure $\langle G \times H, o \rangle$ in which the binary operation 'o' on $G \times H$ is given by $\langle g_1, h_1 \rangle \; o \; \langle g_2, h_2 \rangle = (g_1 * g_2, h_1 \Delta h_2)$ for any $(g_1, h_1), (g_2, h_2) \in G \times H$.

Algebraic structures with two binary operations :-

An algebraic system $\langle S, +, \cdot \rangle$ is called a Ring if the binary operations + and · on S satisfies the following three properties

1) $\langle S, + \rangle$ is an abelian group.
2) $\langle S, \cdot \rangle$ is a semigroup
3) The operation · is distributive over +.
   i.e; for any $a, b, c \in S$, $a \cdot (b+c) = a.b + a.c$ and $(b+c).a = b.a + c.a$.

eg:- 1) Let $R = \{a, b, c, d\}$ and define the operations + & · on R, as shown in the following table then $\langle R, +, \cdot \rangle$ is a Ring.

+	a	b	c	d
a	a	b	c	d
b	b	a	d	c
c	c	d	b	a
d	d	c	a	b

·	a	b	c	d
a	a	a	a	a
b	a	a	b	a
c	a	b	c	d
d	a	a	d	a

2) Let S be a set and P(S) is a power set of S. On P(S) we define operations (+, ·) as follows.

$$A+B = \{x \in S \mid (x \in A \vee x \in B) \wedge (x \notin A \cap B)\}, \; A \cdot B = A \cup B \;\forall$$

$a, b \in P(S)$.

Here $(P(S), +, \cdot)$ is a ring.

* **Subring:-**

A subset $R \subseteq S$, where $\langle S, +, \cdot \rangle$ is a ring, it is called a subring if $\langle R, +, \cdot \rangle$ is itself a ring with the operations + & · restricted to R.

eg 1) The ring of even integers is a subring of the ring of integers.

Sol. Infact if $R \subseteq S$, if we determine that R is closured w.r.to addition and that for any $a \in R$, $-a \in R$ and finally

If R is closed w.r.to operation ·, then R is a sub-string of S. All other properties of Ring are satisfied by R.

For example, the ring of even integers is a subring of integers.

* **Ring Homomorphism:-**

Let $\langle R, +, \cdot \rangle$ and $\langle S, \oplus, \odot \rangle$ be rings. A mapping $g: R \to S$ is called a Ring homomorphism from $\langle R, +, \cdot \rangle$ to $\langle S, \oplus, \odot \rangle$ if for any $a, b \in R$, $g(a+b) = g(a) \oplus g(b)$ and $g(a \cdot b) = g(a) \odot g(b)$

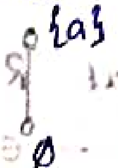
* **Lattice:-**

A lattice is a poset $(L, \leq)$ in which every pair of elements $(a,b) \in L$ has a greatest lower bound and least upper bound.

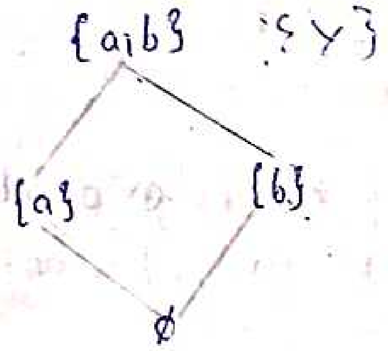
The greatest lower bound of a subset $\{a,b\} \subseteq L$ will be denoted by $a * b$ and the Least upper bound $a \oplus b$.
$a * b$ is also called as Meet (or) Product of a, b. similarly $a \oplus b$ is called Join (or) Sum of a & b. Other symbols such as $\wedge$ and $\vee$ (or) $\cdot$ and $+$ are also used to denote the meet and join of two elements respectively.

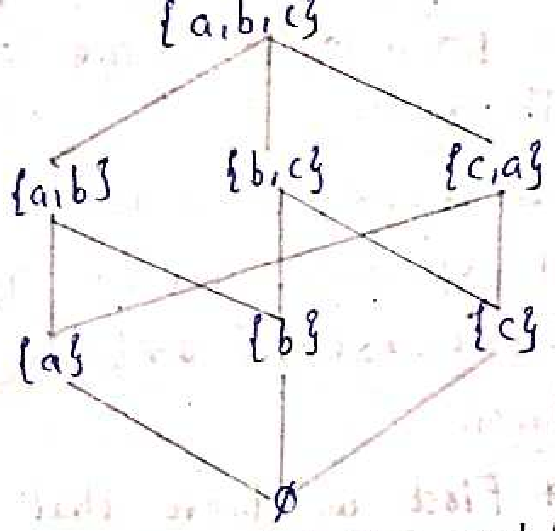
eg:- Let S be any set and P(S) be a power set of S the poset $(P(S), \subseteq)$ is a lattice in which the meet and join are the same as the operations $\cap$ and $\cup$ respectively. In particular when S has a single element the corresponding lattice is a chain containing two elements. when S has two and 3 elements, the diagrams of the corresponding lattices are as shown in the following figures.

$S = \{a\}$



$S = \{a, b\}$ $\quad\quad$ $S = \{a, b, c\}$





2) Let I^+ be the set of all positive integers and let D denote the relation of division in I^+ such that for any $a, b \in I$, $a D b$ iff 'a' divides 'b' then $\langle I^+, D\rangle$ is a lattice in which the join of a & b is given by the least common multiple (LCM) of a & b i.e; $a \oplus b = LCM(a,b)$ and, the meet of a & b i.e; $a * b = GCD(a,b)$.

* Principle of Duality:-

Any statement about lattices involving the operations $*$ and $\oplus$ and the relations $\leq$ and $\geq$ remains true if $*$ is replaced by $\oplus$, $\oplus$ by $*$, $\leq$ by $\geq$ and $\geq$ by $\leq$. The operations $*$ and $\oplus$ are called Duals of each other as are the relations $\leq$ and $\geq$. Similarly the lattices $\langle L, \leq\rangle$ and $\langle L, \geq\rangle$ are called Duals of each other.

Some properties of Lattices:-

Let $\langle L, \leq\rangle$ be a lattice. The two binary operations Meet and Join denote by $*$ and $\oplus$ respectively. For any $a, b, c \in L$, we have

$L-1$: $a * a = a$ $\quad$ $L-1'$: $a \oplus a = a$ [Idempotent]

$L-2$: $a * b = b * a$ $\quad$ $L-2'$: $a \oplus b = b \oplus a$ [Commutative]

$L-3$: $(a * b) * c = a * (b * c)$ $\quad$ $L-3'$: $(a \oplus b) \oplus c = a \oplus (b \oplus c)$ [Associative]

$L-4$: $a * (a \oplus b) = a$ $\quad$ $L-4'$: $a \oplus (a * b) = a$ [Absorption]

Note:- $a * (a \oplus b) = a$

Since we know that $a \leq a$, $a \leq a \oplus b$

$\Rightarrow a \leq a * (a \oplus b) \rightarrow ①$

and $a * (a \oplus b) \leq a \rightarrow ②$

From ① & ② $a * (a \oplus b) = a$

1) Theorem:-

Let $\langle L, \leq \rangle$ be a lattice in which $*$ and $\oplus$ denote the operations of meet and join respectively. For any $a, b \in L$, $a \leq b$ iff $a * b = a \Leftrightarrow a \oplus b = b$

Proof:-

* First we poove that $a \leq b \Leftrightarrow a * b = a$

Let $a \leq b$, we know that $a \leq a$

$\Rightarrow a \leq a * b \rightarrow ①$

Since we know that $a * b \leq a \rightarrow ②$

$\therefore$ From ① & ② $a * b = a$

Hence $a \leq b \Rightarrow a * b = a$

Next assume that $a * b = a$, but it is only possible if $a \leq b$.

i.e; $a * b = a \Rightarrow a \leq b$

Combining these two results we get the required equivalence

i.e; $a \leq b \Leftrightarrow a * b = a$

* It is possible to show that $a \leq b \Leftrightarrow a \oplus b = b$ in a similar manner.

i.e; Let $a \leq b$ and we know that $b \leq b$

Then, $a \oplus b \leq b \rightarrow ①$

since we know that $b \leq a \oplus b \rightarrow ②$

From ① & ② $a \oplus b = b$

Hence $a \leq b \Rightarrow a \oplus b = b$

Next assume that $a \oplus b = b$, but it is possible only if $a \leq b$, Hence $a \oplus b = b \Rightarrow a \leq b$

Combining these two results we get the required equivalence i.e; $a \leq b \Leftrightarrow a \oplus b = b$.

* Next we have to show that $a * b = a \Leftrightarrow a \oplus b = b$.

Let $a*b=a$.

Now $b\oplus(a*b)=b\oplus a=a\oplus b$

and $b\oplus(a*b)=b\oplus(b*a)$

$=b$ ($\because$ By Absorption Law)

$\therefore a\oplus b=b$

Conversely suppose that $a\oplus b=b$

Now $a*(a\oplus b)=a*b$

and $a*(a\oplus b)=a$ [$\because$ By absorption Law]

$\therefore a*b=a$

Hence $a*b=a \Leftrightarrow a\oplus b=b$.

2) Let $\langle L,\leq\rangle$ be a lattice for any $a,b,c\in L$ the following properties called Isotonicity hold, $b\leq c \Rightarrow \begin{cases} a*b\leq a*c \\ a\oplus b\leq a\oplus c \end{cases}$

Proof:-

Since we know that $b\leq c \Leftrightarrow b*c=b \Leftrightarrow b\oplus c=c$

To show that $a*b\leq a*c$, we have to show that $(a*b)*(a*c)=(a*b)$

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

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

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

$=a*b$

$\therefore a*b\leq a*c$

Similarly to show that $a\oplus b\leq a\oplus c$, we shall show that $(a\oplus b)\oplus(a\oplus c)=(a\oplus c)$

Now $(a\oplus b)\oplus(a\oplus c)=((a\oplus a)\oplus(b\oplus c))$

$=a\oplus(b\oplus c)$ ($\because b\oplus c=c$)

$=a\oplus c$

$\therefore a\oplus b\leq a\oplus c$

Hence, the theorem.

3) Let $\langle L,\leq\rangle$ be a lattice. For any $a,b,c\in L$. The following inequalities called Distributive inequalities hold $a\oplus(b*c)\leq(a\oplus b)*(a\oplus c)$ and $a*(b\oplus c)\geq(a*b)\oplus(a*c)$

Proof:- Since we know that $a\leq a\oplus b$ and $a\leq a\oplus c$

$\Rightarrow a\leq(a\oplus b)*(a\oplus c) \rightarrow (1)$

Since $(b*c)\leq b\leq(a\oplus b) \rightarrow (2)$

Since $(b*c) \le c \le (a \oplus c)$ → (3)

From (2) & (3), $(b*c) \le (a\oplus b) * (a \oplus c)$ → (4)

From (1) & (4), $(a \oplus (b*c)) \le (a\oplus b) * (a\oplus c)$

Similarly for the second condition.

Since we know that $a \ge a*b$ and $a \ge a*c$

$\Rightarrow a \ge (a*b) \oplus (a*c)$ → (5)

Since $b\oplus c \ge b \ge (a*b)$ → (6)

Since $b \oplus c \ge c \ge (a*c)$ → (7)

From (6) & (7), $b\oplus c \ge (a*b) \oplus (a*c)$ → (8)

From (5) & (8) $a*(b\oplus c) \ge (a*b) \oplus (a*c)$

4) Let $\langle L, \le \rangle$ be a lattice for any $a,b,c \in L$. The following Modular inequality $a \le c \Leftrightarrow a \oplus (b*c) \le (a\oplus b) * c$

Proof:- Let $a \le c$ then $a \oplus c = c$

Since we know that $a \oplus (b*c) \le (a\oplus b) * (a \oplus c)$

Substitute $a \oplus c = c$ then $a \oplus (b*c) \le (a\oplus b) * c$

Conversely suppose that $a \oplus (b*c) \le (a \oplus b) * c$

Since $a \le a \oplus (b*c)$ and $(a\oplus b)*c \le c$

$\therefore a \le a \oplus (b*c) \le (a\oplus b) * c \le c$

$\therefore a \le c$

5) Show that in a lattice if $a \le b \le c$ then $a\oplus b = b*c$ and $(a*b) \oplus (b*c) = b = (a \oplus b) * (a \oplus c)$

Proof:- Since $a \le b \le c$

i.e; if $a \le b \Rightarrow a \oplus b = b$, $a*b = a$

if $b \le c \Rightarrow b*c = b$, $b \oplus c = c$

$\therefore$ If $a \le b \le c \Rightarrow a \oplus b = b*c$

Now $(a*b) \oplus (b*c) = a \oplus b = b$ $(\because a*b = a, b*c = b)$

and $(a\oplus b) * (a \oplus c) = b * c = b$ $(\because a \le c, (a \oplus c) = c)$

$\therefore (a*b) \oplus (b*c) = b = (a\oplus b) * (a \oplus c)$

Show that in a lattice if $a \le b$ and $c \le d$ then $a*c \le b*d$

Since $a \le b \Leftrightarrow a*b = a \Leftrightarrow a \oplus b = b$

$c \leq d \Leftrightarrow c * d = c \Leftrightarrow c \oplus d = d$

To show that $a*c \leq b*d$, we will prove

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

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

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

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

$= [(a*b)*(c*d)]$ (Associative)

$= a*c$

$\therefore a*c \leq b*d$

* **Lattices as Algebraic systems:-**

A lattice is an algebraic system $\langle L, *, \oplus \rangle$ with two binary operations $*$ and $\oplus$ on L which are both

1: Commutative

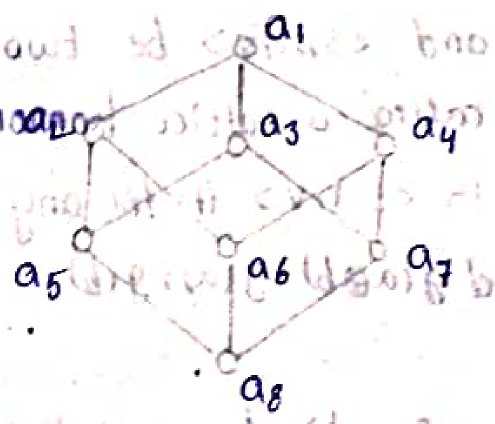
2: Associative

3: Absorption Laws

* **Sub Lattice:-**

Let $\langle L, *, \oplus \rangle$ be a lattice and $S \subseteq L$ be a subset of L. The algebra $\langle S, *, \oplus \rangle$ is a sub lattice of $\langle L, *, \oplus \rangle$ if and only if S is closed under both operations $*$ & $\oplus$

Eg:- Let $\langle L, \leq \rangle$ be a lattice with $L = \{a_1, a_2, \cdots a_8\}$ and S_1 and S_2 and S_3 be subsets of L given by $S_1 = \{a_1, a_2, a_4, a_6\}$, $S_2 = \{a_3, a_5, a_7, a_8\}$, $S_3 = \{a_1, a_2, a_4, a_8\}$. The diagram of $\langle L, \leq \rangle$ is given in the following figure. observe that $\langle S_1, \leq \rangle$ and $\langle S_2, \leq \rangle$ are sublattices of $\langle L, \leq \rangle$ but $\langle S_3, \leq \rangle$ is not a sublattice because $a_2, a_4 \in S_3$ but $a_2 * a_4 = a_6 \notin S_3$.



* Direct Product:-

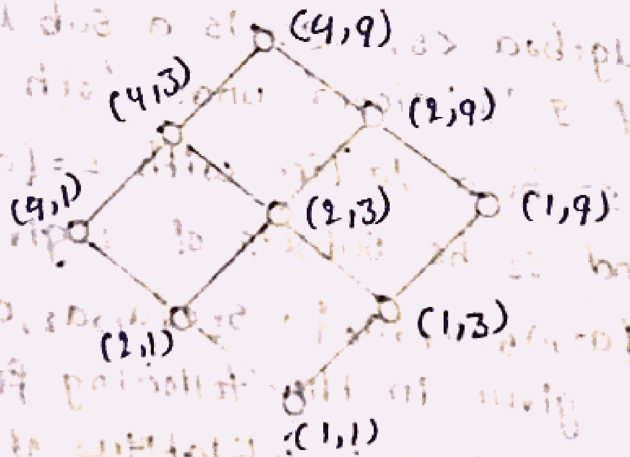
Let $\langle L, *, \oplus \rangle$ and $\langle S, \wedge, \vee \rangle$ be two lattices. The algebraic system $\langle L \times S, \cdot, + \rangle$ in which the binary operations $+$ and $\cdot$ on $L \times S$ are such that for any $\langle a_1, b_1 \rangle$ and $\langle a_2, b_2 \rangle$ in $L \times S$ then

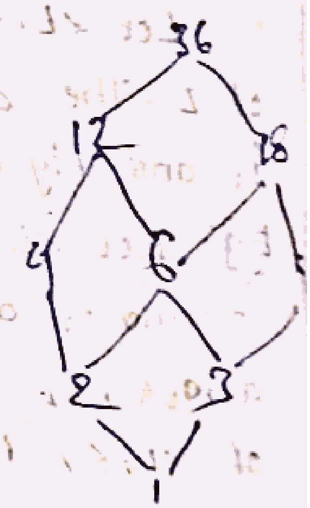
$$\langle a_1, b_1 \rangle \cdot \langle a_2, b_2 \rangle = \langle a_1 * a_2, b_1 \wedge b_2 \rangle$$

$$\langle a_1, b_1 \rangle + \langle a_2, b_2 \rangle = \langle a_1 \oplus a_2, b_1 \vee b_2 \rangle$$

is called the direct product of the lattices $\langle L, *, \oplus \rangle$ and $\langle S, \wedge, \vee \rangle$

Eg:- Consider the chains of divisors of 4 & 9 i.e; $L_1 = \{1, 2, 4\}$ and $L_2 = \{1, 3, 9\}$ and the partial ordering relation of division on L_1 and L_2. the lattice $L_1 \times L_2$ is shown in the following figure. Notice that the diagram of the latice of divisors of 36 is the same as the above Hasse diagram. Except that the node $\langle a, b \rangle$ is replaced by a product ab.





* Lattice Homomorphism:-

Let $\langle L, *, \oplus \rangle$ and $\langle S, \wedge, \vee \rangle$ be two lattices. A mapping $g: L \to S$ is called a lattice homomorphism from the lattice $\langle L, *, \oplus \rangle$ to $\langle S, \wedge, \vee \rangle$ if for any $a, b \in L$, $g(a * b) = g(a) \wedge g(b)$ and $g(a \oplus b) = g(a) \vee g(b)$

* Order-Preserving:-

Let $\langle P, \leq \rangle$ and $\langle Q, \leq' \rangle$ be two partially ordered sets. A mapping $f: P \to Q$ is said to be order-preserving relative to the ordering $\le$ in P and $\le'$ in Q iff for any $a, b \in P$ such that $a \le b$, $f(a) \le' f(b)$ in Q.

* **Order-Isomorphic:-**

Two partially ordered sets $\langle P, \le \rangle$ and $\langle Q, \le' \rangle$ are called order isomorphic if there exists a mapping $f: P \to Q$ which is bijective and if both f and f^{-1} are order-preserving.

* **Complete lattice:-**

A lattice is called complete if each of its non empty subsets has a least upperbound and a greatest lowerbound.

The least and greatest elements of a lattice if they exist are called the bounds of the lattice and are denoted by 0 and 1 respectively. A lattice which has both elements 0 and 1 is called a bounded lattice for the lattice $\langle L, *, \oplus \rangle$ with $L = \{a_1, a_2, \ldots, a_n\}$ $\overset{n}{\underset{i=1}{*}} a_i = 0$ and $\overset{n}{\underset{i=1}{\oplus}} a_i = 1$. The bounds 0 and 1 of a lattice $\langle L, *, \oplus, 0, 1 \rangle$ satisfy the following identities. for any $a \in L$

$a \oplus 0 = a \qquad a * 1 = a$

$a \oplus 1 = 1 \qquad a * 0 = 0$

* **Complemented Lattice:-**

In a bounded lattice $\langle L, *, \oplus, 0, 1 \rangle$ an element $b \in L$ is called a complement of an element $a \in L$ if $a * b = 0$ and $a \oplus b = 1$

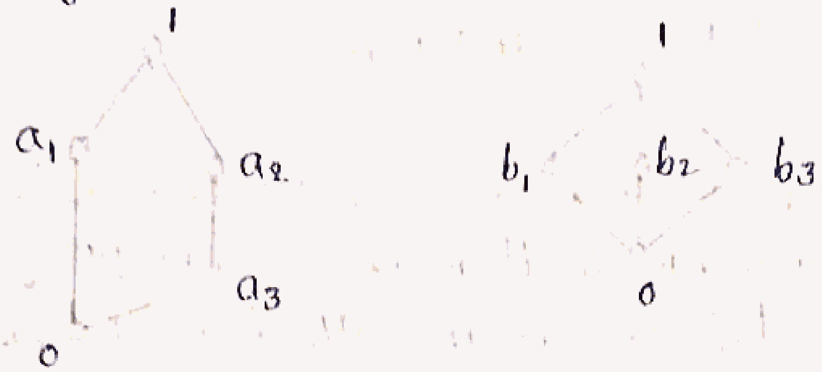
A lattice $\langle L, *, \oplus, 0, 1 \rangle$ is said to be a "complemented lattice" if every element of L has atleast one.

* **Distributive Lattice:-**

A lattice $\langle L, *, \oplus \rangle$ is called a Distributive lattice if for any $a, b, c \in L$, $a * (b \oplus c) = (a * b) \oplus (a * c)$

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

Eg:- Show that the lattices given in the following diagram are not distributive.



fig(a)

In fig(a), $a_2 * (a_1 \oplus a_3) = a_2 * 1 = a_2$ but $(a_2 * a_1) \oplus (a_2 * a_3)$ $= 0 \oplus a_3 = a_3$

$\therefore$ Fig(a) is not a distributive lattice

Where as in fig(b), $b_1 * (b_2 \oplus b_3) = b_1 * 1 = b_1$ but $(b_1 * b_2) \oplus (b_1 * b_3) = 0 \oplus 0 = 0$

$\therefore$ fig(b) is not a distributive lattice.

* **Theorems:-**

1) Every chain is a Distributive lattice.

Proof:-

Let $\langle L, \leq \rangle$ be a chain and $a, b, c \in L$

consider the following cases.

(i) $a \leq b$ and $a \leq c$ (ii) $a \geq b$ and $a \geq c$

We shall now show that the distributive law is satisfied by a, b, c

For case (i) $a * (b \oplus c)$

Since $a \leq b$ and $a \leq c \Rightarrow a \leq b \oplus c$

Then we know that $a * (b \oplus c) = a \rightarrow ①$

Now $(a * b) \oplus (a * c)$

Since $a \leq b$ then $a * b = a$

$a \leq c$ then $a * c = a$

$(a * b) \oplus (a * c) = a \rightarrow ②$

From ① & ②

$a * (b \oplus c) = (a * b) \oplus (a * c)$.

Similarly we prove that $a \oplus (b * c) = (a \oplus b) * (a \oplus c)$

$\therefore$ L is a distributive lattice

$\therefore$ Every chain is a distributive lattice.

2) Let $\langle L, *, \oplus \rangle$ be a distributive lattice. For any $a, b \in L$

$((a*b) = a*c) \wedge (a \oplus b = a \oplus c) \Rightarrow b = c$

Proof:- Since 'L' is a distributive lattice

consider $(a*b) \oplus c = (a*c) \oplus c = c$

Now $(a*b) \oplus c = (a \oplus c) * (b \oplus c)$

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

$= (a * c) \oplus b$

$= (a * b) \oplus c$

$= b$

$\therefore b = c$

3) Show that Demorgan's law given by $(a*b)' = a' \oplus b'$ and $(a \oplus b)' = a' * b'$ holds in a complemented, distributive lattice.

Proof:- Inorder to show that $(a*b)' = a' \oplus b'$

we will prove $(a*b) * (a' \oplus b') = 0$ as well as $(a*b) \oplus (a' \oplus b') = 1$

Now $(a*b) * (a' \oplus b') = [(a*b)*a'] \oplus [(a*b)*b']$ [It is distributive lattice]

$= [(b*a)*a'] \oplus [(a*b)*b']$ [Commutative]

$= [b*(a*a')] \oplus [a*(b*b')]$ [Associative]

$= [b*0] \oplus [a*0]$

$= 0 \oplus 0 = 0$

Similarly $(a*b) \oplus (a' \oplus b')$

Since $(a*b) \oplus (a' \oplus b')$

Since the given lattice is distributive

$= [a \oplus (a' \oplus b')] * [b \oplus (a' \oplus b')]$

$= [(a \oplus a') \oplus b'] * [(b \oplus b') \oplus a']$

$= [1 \oplus b'] * [1 \oplus a']$

$= 1 * 1 = 1$

$\therefore (a*b)' = a' \oplus b'$

Similarly we can prove that $(a \oplus b)' = a' * b'$

* **Boolean Algebra:-**

A boolean algebra is a complemented, distributive lattice. Generally it is denoted by $\langle B, *, \oplus, ', 0, 1\rangle$ in which $\langle B, *, \oplus\rangle$ is a lattice with 2 binary operations $*$ & $\oplus$ called the Meet and Join respectively. The corresponding poset will be denoted by $\langle B, \leq\rangle$. The bounds of the lattice are denoted by 0 and 1, where 0 is the least element and 1 is the greatest element of $\langle B, \leq\rangle$.

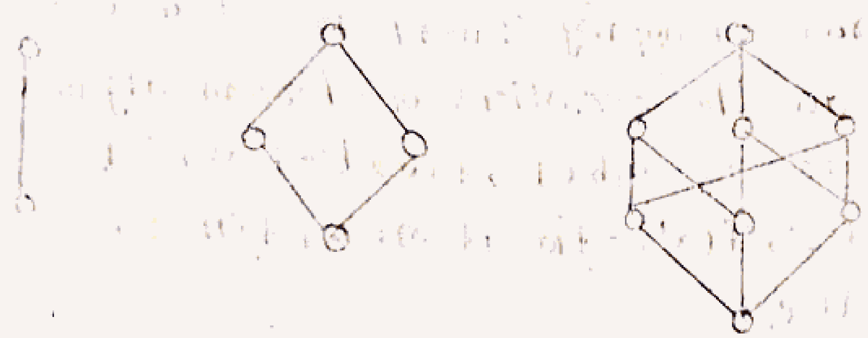
Since $\langle B, *, \oplus\rangle$ is complemented and because of the fact that it is distributive lattice, each element of B has a unique complement. we shall denote the unary operation of complementation by ' ' ', so that for any $a \in B$, the complement of 'a' is denoted by $a' \in B$.

eg:-1) Let $B = \{0, 1\}$ be a set. The operations $*$, $\oplus$ and ' are on B are given in the following table.

$*$	0	1	$\oplus$	0	1	x	x'
0	0	0	0	0	1	0	1
1	0	1	1	1	1	1	0

The algebra $\langle B, *, \oplus, ', 0, 1\rangle$ satisfies all the properties of complemented and distributive lattices. It is a two element boolean algebra.

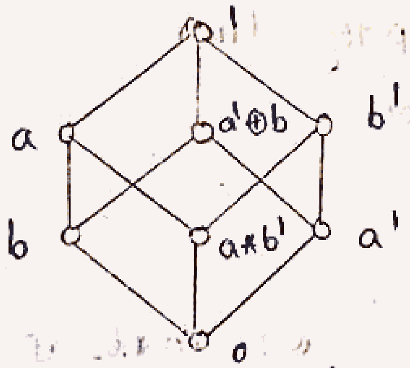
2) Let S be a non empty set and P(S) be the powerset. The set algebra $\langle P(S), \cap, \cup, \sim, \phi, S\rangle$ is a boolean algebra in which the complement of any subset $A \subseteq S$ is $\sim A = S - A$. If S has n elements then P(S) has 2^n elements the diagrams for the boolean algebra when S has 1,2,3 elements are given in the following figure. If S is an empty set then P(S) has only one element, i.e; ϕ is a boolean algebra so that $\phi = 0 = 1$ and the corresponding boolean algebra



* **Sub Algebra:-**

Let $\langle B, *, \oplus, ', 0, 1\rangle$ be a boolean algebra and $S \subseteq B$. If S contains the elements 0 and 1 and is closed under the operations $*$, $\oplus$ and $'$, then $\langle S, *, \oplus, ', 0, 1\rangle$ is called a "sub boolean algebra".

eg:- 1) consider the boolean algebra given in the following figure. Let subsets be $S_1 = \{a, a', 0, 1\}$, $S_2 = \{a'\oplus b, a*b', 0, 1\}$, $S_3 = \{a*b', b', a, 1\}$, $S_4 = \{b', a*b', a', 0\}$, $S_5 = \{a, b', 0, 1\}$



The subsets S_1 and S_2 are sub boolean algebras

* **Direct Product:-**

Let $\langle B_1, *_1, \oplus_1, ', 0_1, 1_1\rangle$ and $\langle B_2, *_2, \oplus_2, '', 0_2, 1_2\rangle$ be two boolean algebras. The direct product of these two is defined to be a boolean algebra that is given by $\langle B_1 \times B_2, *_3, \oplus_3, ''', 0_3, 1_3\rangle$ in which the operations are defined for any $(\langle a_1, b_1\rangle$ and $\langle a_2, b_2\rangle) \in B_1 \times B_2$ as

$\langle a_1, b_1\rangle *_3 \langle a_2, b_2\rangle = \langle a_1 *_1 a_2, b_1 *_2 b_2\rangle$.

$\langle a_1, b_1\rangle \oplus_3 \langle a_2, b_2\rangle = \langle a_1 \oplus_1 a_2, b_1 \oplus_2 b_2\rangle$.

$\langle a_1, b_1\rangle''' = \langle a_1', b_1''\rangle$

$0_3 = \langle 0_1, 0_2\rangle$, $1_3 = \langle 1_1, 1_2\rangle$

* **Boolean homomorphism:-**

Let $\langle B, *, \oplus, ', 0, 1\rangle$ and $\langle P, \cap, \cup, -, \alpha, \beta\rangle$ be two boolean algebras. A mapping $f: B \to P$ is called a "Boolean homomorphism" if all the operations of boolean algebra are preserved i.e; for any $a, b \in B$, $f(a * b) = f(a) \cap f(b)$, $f(a \oplus b) = f(a) \cup f(b)$, $f(a') = \overline{f(a)}$, $f(0) = \alpha$, $f(1) = \beta$.

* Join-irreducible:-

Let $\langle L, *, \oplus \rangle$ be a lattice. An element $a \in L$ is called join-irreducible if it cannot be expressed as the join of two distinct elements of L. In otherwords $a \in L$ is join-irreducible if for any $a_1, a_2 \in L$, $a = a_1 \oplus a_2 \Rightarrow (a = a_1)$ or $(a = a_2)$.

1) Prove the following boolean identities.

(a) $a \oplus (a' * b) = a \oplus b$

$L.H.S = a \oplus (a' * b)$

Apply distributive property then

$= (a \oplus a') * (a \oplus b)$

$= 1 * (a \oplus b)$

$= a \oplus b$

$= R.H.S$

$*$ means lower bound
$\oplus$ upper bound

(b) $a * (a' \oplus b) = a * b$

$L.H.S = a * (a' \oplus b)$

$= (a * a') \oplus (a * b)$

$= 0 \oplus (a * b)$

$= a * b = R.H.S$

(c) $(a * b) \oplus (a * b') = a$

$L.H.S = (a * b) \oplus (a * b')$

$= a * (b \oplus b')$

$= a * 1$

$= a = R.H.S$

(d) $(a * b * c) \oplus (a * b) = a * b$

$L.H.S = (a * b * c) \oplus (a * b)$

$= ((a * b) * c) \oplus (a * b)$ [Absorption law]

$= a * b = R.H.S$

2) In any boolean algebra, S.T. $a \le b \Rightarrow a + bc = b(a + c)$

Sol. Let $a \le b$

then we know that $a * b = a$ and $a \oplus b = b$

Now $a + bc = ab + ac$ $(\because a = a * b)$

$= b(a + c)$

3) Simplify the following boolean expressions.
$(a' * b' * c) \oplus (a * b' * c) \oplus (a * b' * c')$

Sol: $(a' * b' * c) \oplus (a * b' * c) \oplus (a * b' * c')$

$= [(a' \oplus a) * (b' * c)] \oplus (a * b' * c')$

$= [1 * (b' * c)] \oplus (a * b' * c')$

$= (b' * c) \oplus (a * b' * c')$

$= b' * (c \oplus (a * c')) = b' * ((c \oplus a) * (c \oplus c'))$

$= b' * (c \oplus a)$

* **Boolean Expression:-**

A boolean expression, form or formula in n variables $x_1, x_2, \ldots x_n$ is any finite string of symbols formed in the following manner.

1) 0 & 1 are boolean expressions.

2) $x_1, x_2, \ldots x_n$ are boolean expressions

3) If α_1 & α_2 are boolean expressions then $\alpha_1 * \alpha_2$ and $\alpha_1 \oplus \alpha_2$ are also boolean expressions.

4) If α is a boolean expression then $(\alpha)'$ is also a boolean expression.

5) No strings of symbols except those formed in accordance with rules 1, to 4 are boolean expressions.

* **Equivalence of Boolean expressions:-**

Two boolean forms $\alpha(x_1, x_2, \ldots x_n)$ and $\beta(x_1, x_2, \ldots x_n)$ are called equivalent if one can be obtained from the other by a finite no. of applications of the identities of a boolean algebra.

* **Minterm:-**

A boolean form in n variables $x_1, x_2, \ldots x_n$ consisting of the product of n terms such as $x_1^{a_1} * x_2^{a_2} * \ldots * x_n^{a_n} = \overset{n}{\underset{i=1}{*}} x_i^{a_i}$ in which a_i is either 0 or 1, x_i^0 stands for x_i' and x_i^1 stands for x_i is called a Minterm, complete product, or a fundamental product of the n variables.

1) Write the following boolean expressions in an equivalent sum of products canonical form, in three variables x_1, x_2 & x_3.

(a) $x_1 * x_2 = (x_1 * x_2) * (x_3 \oplus x_3') = (x_1 * x_2 * x_3) \oplus (x_1 * x_2 * x_3')$

Sol. (a) $x_1 * x_2 = x_1 * x_2 * (x_3 \oplus x_3')$

$= (x_1 * x_2 * x_3) \oplus (x_1 * x_2 * x_3')$

$= m_7 \oplus m_6$

$x=1, x'=0$
000 — m_0
001 — m_1
010 — m_2
011 — 3 — m_3
100 — m_4
101 — m_5
110 — m_6
111 — m_7

(b) $x_1 \oplus x_2$

$x_1 \oplus x_2 = [x_1 * (x_2 \oplus x_2')] \oplus [x_2 * (x_1 \oplus x_1')]$

$= [(x_1 * x_2) \oplus (x_1 * x_2')] \oplus [(x_2 * x_1) \oplus (x_2 * x_1')]$

$= [(x_1 * x_2) * (x_3 \oplus x_3')] \oplus [(x_1 * x_2') * (x_3 \oplus x_3')]$

$\oplus [(x_2 * x_1) * (x_3 \oplus x_3')] \oplus [(x_2 * x_1') * (x_3 \oplus x_3')]$

$= (x_1 * x_2 * x_3) \oplus (x_1 * x_2 * x_3') \oplus (x_1 * x_2' * x_3) \oplus (x_1 * x_2' * x_3')$

$\oplus (x_1 * x_2 * x_3) \oplus (x_1 * x_2 * x_3') \oplus (x_1' * x_2 * x_3) \oplus (x_1' * x_2 * x_3')$

$= m_7 \oplus m_6 \oplus m_5 \oplus m_4 \oplus m_3 \oplus m_2$

Note:- Every boolean expression consisting of n-variables is equivalent to a boolean expression consisting of the product of Max terms only. Such a canonical form is known as the product of sums canonical form.

eg:- Obtain the product of sums canonical form of the boolean expression of $x_1 * x_2$

$x=0, x'=1$
00 — max_0
01 — max_1
10 — max_2
11 — max_3

Sol. (a) $x_1 * x_2 = [x_1 \oplus (x_2 * x_2')] * [(x_1 * x_1') \oplus x_2]$

$= (x_1 \oplus x_2) * (x_1 \oplus x_2') * (x_1 \oplus x_2) * (x_1' \oplus x_2)$

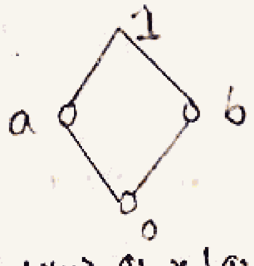
$= max_0 * max_1 * max_2$

* Values of Boolean expressions and boolean functions:-

Let $\alpha(x_1, x_2, \ldots x_n)$ be a boolean expression in n-variables and $\langle B, *, \oplus, ', 0, 1 \rangle$ be any boolean algebra whose elements are denoted by $a_1, a_2, \ldots$. Let $\langle a_1, a_2, \ldots a_n \rangle$ be an n-tuple of B^n. If we replace x_1 by a_1 and x_2 by a_2 ...... and x_n by a_n in the boolean expression $\alpha(x_1, x_2, \ldots x_n)$, we obtain an expression which represents an element of B. We shall denote the result of expression by $\alpha(a_1, a_2, \ldots a_n) \in B$ and call it the value of the boolean expression, $\alpha(x_1, x_2, \ldots x_n)$, for the n-tuple $\langle a_1, a_2, \ldots a_n \rangle \in B^n$

it is possible to determine the values of the boolean expression $\alpha(x_1, x_2, \dots x_n)$ for every n-tuple of B^n. The process of determining all such values is called a valuation process over the boolean algebra $\langle B, *, \oplus, ', 0, 1\rangle$.

1) Find the value of $x_1 * x_2 * [(x_1 * x_4) \oplus x_2' \oplus (x_3 * x_1')]$ for $x_1 = a$, $x_2 = 1$, $x_3 = b$, $x_4 = 1$ where $a, b, 1 \in B$ and the boolean algebra $\langle B, *, \oplus, ', 0, 1\rangle$ is shown in the following figure.



Sol. $= x_1 * x_2 * ((x_1 * x_4) \oplus x_2' \oplus (x_3 * x_1'))$

$\Rightarrow a * 1 * [(a * 1) \oplus 0 \oplus (b * a')]$

$\Rightarrow a * [a \oplus (b * a')]$

$\Rightarrow a * [(a \oplus b) * (a \oplus a')]$

$\Rightarrow a * [(a \oplus b) * 1]$

$\Rightarrow a * (a \oplus b)$ (By Absorption Law)

$\Rightarrow a$

* Boolean Function:-

Let $\langle B, *, \oplus, ', 0, 1\rangle$ be a boolean algebra. A function $f: B^n \to B$ is associated with a boolean expression in n variables is called a "boolean function."

A boolean expression in n variables $x_1, x_2, \dots x_n$ is called symmetric if interchanging any two variables results in an equivalent expression.

eg:- $(x_1 * x_2') \oplus (x_1' * x_2)$ which is equal to $(x_2 * x_1) \oplus (x_2' * x_1)$

Unit - 3

Graphs And Trees

Graph Theory :-

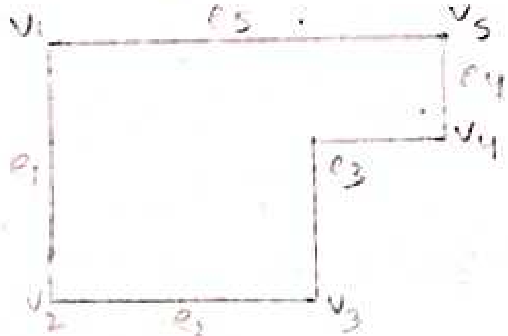
Graph :

A Graph can be represented as $G(V,E)$ where 'V' represents set of vertices in the graph and 'E' is a set of edges those connect the vertices of G.

Simple Graph :

A Simple Graph $G(V,E)$ consists of V, a non-empty set of vertices and E a set of unordered pairs of distinct elements of V called edges.

Eg:



$V = \{v_1, v_2, v_3, v_4, v_5\}$

$E = \{e_1, e_2, e_3, e_4, e_5\}$

Multi Graph :

A Graph $G(V,E)$ consists of set V of vertices, a set E of edges and function f from E to $\{\{u,v\} / u,v \in V$ and $u \neq v\}$. The edges e_1 and e_2 are called multiple (or) Parallel edges.

Eg:

Pseudo Graph :-

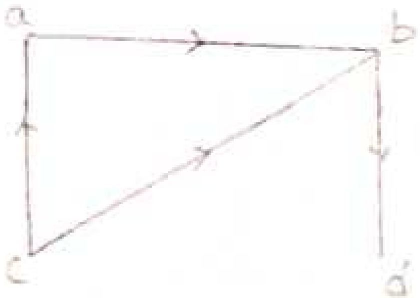
A Pseudo Graph $G = (V, E)$ contains of a set V of vertices and a set E of edges and a function f from E to $\{\{u,v\} / u,v \in V\}$

An edge is called a loop if there exists an edge in between a vertex and itself i.e., $f(e) = \{u,v\} = \{u\}$ for some $u \in V$

Directed Graph :-

A Directed Graph $G = (V, E)$ Consists of a set of vertices V and a set of edges E that are ordered pairs of the elements of V i.e., every edge of a graph is directed edge.

Eg:



Adjacent vertices :-

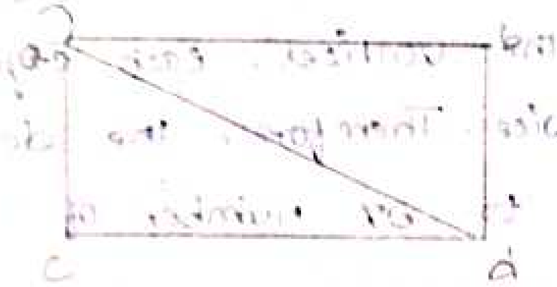
Two vertices u and v in an undirected graph G are called "Adjacent". If $\{u,v\}$ is an edge of G If $e = \{u,v\}$ then e is called incident with the vertices u and v, the vertices u and v are called end points of the edge.

When (u,v) is an edge of the graph G with directed edges, u is said to be adjacent to v and v is said to be adjacent from u. The vertex u is called the Initial vertex and v is called Terminal vertex of the edge

Degree of a vertex :-

The degree of a vertex in an undirected graph is the number of edges incident with it, except that a loop at a vertex is counted twice to the degree of that vertex. It is denoted by deg(v).

Eg:



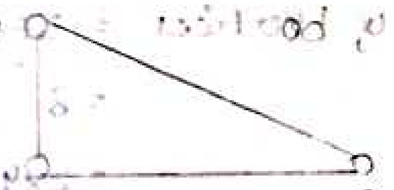
deg(a) = 5 , deg(c) = 3

deg(b) = 2 , deg(d) = 2.

Isolated Vertex :-

A vertex of degree zero is called Isolated vertex.

Eg:

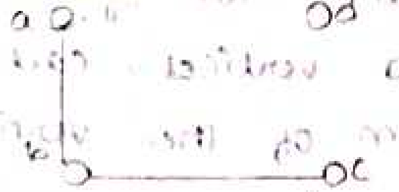


Here, d is an Isolated vertex

Pandant vertex :-

A vertex of degree '1' is called Pandant vertex.

Eg:



Here ; a and c are called Pendant vertices.

Hand Shaking Theorem :-

Let $G(V,E)$ be an undirected graph with e number of edges then $\sum_{v \in V} deg(v) = 2e$, (i.e.), the degree sum of the vertices of the given undirected graph is equal to twice the number of its edges.

This is true because while counting the degree sum of the vertices, each edge is exactly counted twice. Therefore, the degree sum of vertices equal to 2x number of edges

Eg:

$deg(a) = 3$, $deg(c) = 2$

$deg(b) = 2$, $deg(d) = 1$

Now, degree sum of vertices $= 3+2+2+1$

$= 8$

$= 2 \times 4$ (no. of edges)

$\therefore$ degree sum of vertices $= 2 \times$ no. of edges.

Problem :-

1) How many edges are there in the graph with 10 vertices and each of degree 6.

Sol: Since, there are 10 vertices, each of degree 6 then degree sum of the vertices $= 10 \times 6$

$= 60$

By Hand Shaking theorem it follows that

$$2e = 60$$
$$\Rightarrow e = 60/2$$
$$\Rightarrow e = 30$$

$\therefore$ No. of Edges $= 30$

Theorem :-

An undirected graph has an even number of vertices of odd degree.

Proof :

Let V_1 and V_2 be the set of vertices of even degree and odd degree respectively in an undirected graph. Then

$$\sum_{v \in V} deg(v) = \sum_{v \in V_1} deg(v) + \sum_{v \in V_2} deg(v)$$

By Hand Shaking theorem, we can write

$$2e = \sum_{v \in V_1} deg(v) + \sum_{v \in V_2} deg(v)$$

$$\Rightarrow \sum_{v \in V_1} deg(v) + \sum_{v \in V_2} deg(v) = \text{Even} \quad (\because \sum_{v \in V} deg(v) = 2)$$

$$\Rightarrow \text{Even number} + \sum_{v \in V_2} deg(v) = \text{Even number}$$

$$\Rightarrow \sum_{v \in V_2} deg(v) = \text{Even number} - \text{even number}$$
$$= \text{even}$$

Since, the sum of the degree of the degree vertices is even, then those odd vertices must be even number then only condition will be satisfied. Therefore, an undirected graph as an Even number of odd degree vertices.

Indegree and outdegree of a vertex :-

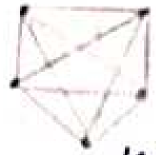
In a directed graph the Indegree of a vertex v is the number of edges with v as their terminal vertex. The outdegree of v denoted by $deg^+(v)$ is the Number of edges with v as their initial vertex. A vertex with in degree zero can be called as source and a vertex with outdegree zero can be called as sink.

Indegree	Outdegree
$deg^-(v_1) = 0$	$deg^+(v_1) = 3$
$deg^-(v_2) = 1$	$deg^+(v_2) = 1$
$deg^-(v_3) = 3$	$deg^+(v_3) = 0$
$deg^-(v_4) = 1$	$deg^+(v_4) = 1$

Here v_1 is source and v_4 is sink.

Complete Graph :

A graph is said to be complete if there exist exactly one edge between each and every pair of distinct vertices in the graph. Complete graph is denoted by 'K'.

K_1 K_2 K_3 K_4 K_5

Cycle :-

The cycle C_n, $n \geq 3$ consists of n vertices v_1, v_2 and v_n and edges $\{v_1, v_2\}, \{v_2, v_3\} \ldots \{v_{n-1}, v_n\}$, $\{v_n, v_1\}$.

Eg:

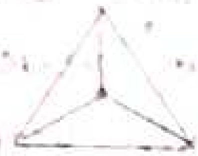
 

C_3 C_4

Wheel :-

We obtain wheel W_n when we had an additional vertex to the cycle C_n, $n \geq 3$ and connect this new vertex to each of the n vertices in C_n by new edges

Eg:

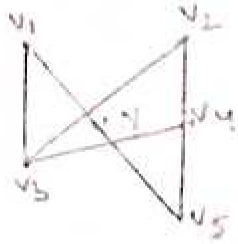
 

W_3 W_4

Bipartite Graph :-

A simple graph G is called Bipartite if its vertex set V can be partitioned into two disjoint subsets V_1 and V_2 such that every edge in the graph connects a vertex in V_1 and a vertex in V_2 i.e., no edge in G connects either two vertices in V_1 or two vertices in V_2

Eg:



Here we can partition the vertex set into two sets V_1 and V_2 as $V_1 = \{v_1, v_2, v_4\}$, $V_2 = \{v_3, v_5\}$ there are no edges in between the vertices in of V_1 as well as the vertices in V_2 and the edges are in between the vertices of V_1 and the vertices of V_2. Hence it is a Bipartite graph

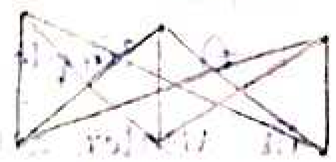
Complete Bipartite graph:

The Complete Bipartite graph K_{mn} is the graph is whose vertex set is partition into two subsets of small m and n vertices respectively. There is an edge between two vertices if and only iff one vertex is in the first subset and the other vertex is in the second subset.

Eg: 1)

$K_{2,3}$

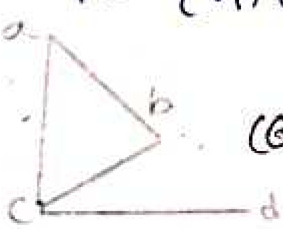
2)



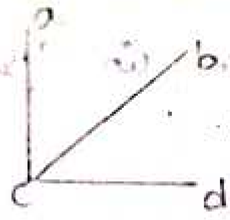
$K_{3,3}$

SubGraph:-

A subgraph of a Graph $G = (V, E)$ is a graph $H = (V_1, E_1)$ where $V_1 \subseteq V$ and $E_1 \subseteq E$



(G)



(H)

Adjacency Matrix :-

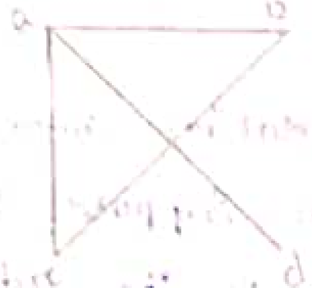
It gives the Adjacency relation among the vertices of the given graph. Suppose $G = (V, E)$ is a simple graph and $|V| = n$, the adjacency matrix A of G is $n \times n$ matrix, whose $(i, j)^{th}$ entry is '1', if there is an edge between the vertices v_i and v_j, otherwise '0' i.e.,

$$a_{ij} = \begin{cases} 1 & \text{if } \{v_i, v_j\} \text{ is an edge of } G \\ 0 & \text{otherwise} \end{cases}$$

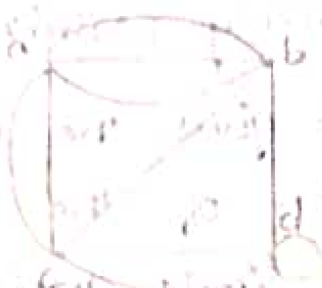
Adjacency matrix of a simple graph is symmetric and $a_{ii} = 0$ since a simple graph has no loops. Adjacency matrix can also be used to represent undirectory graph with loops and multiple edges. A loop at a vertex a_i is represented by a_{ii} at $(i, i)^{th}$ position of the adjacency matrix.

When multiple edges are present, the adjacency matrix is no longer a zero-one matrix. Since $(i, j)^{th}$ entry of this matrix is equal to the number of edges that are associated to $\{a_i, a_j\}$

Similarly the matrix where a directed graph $G = (V, E)$ has a '1' in its $(i, j)^{th}$ position, if there is an edge from (v_i, v_j). The Adjacency matrix of a directed graph don't have to be symmetric.



G_1



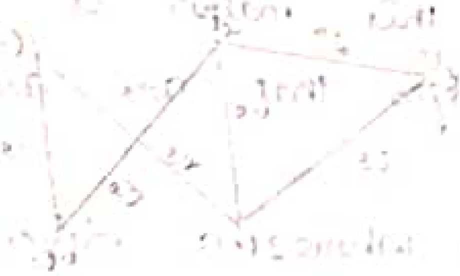
G_2

$$A_{G_1} = \begin{array}{c} \\ a \\ b \\ c \\ d \end{array} \begin{array}{c} \begin{array}{cccc} a & b & c & d \end{array} \\ \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \end{array} \qquad A_{G_2} = \begin{array}{c} \\ a \\ b \\ c \\ d \end{array} \begin{array}{c} \begin{array}{cccc} a & b & c & d \end{array} \\ \begin{bmatrix} 0 & 3 & 0 & 2 \\ 3 & 0 & 1 & 1 \\ 0 & 1 & 1 & 2 \\ 2 & 1 & 2 & 0 \end{bmatrix} \end{array}$$

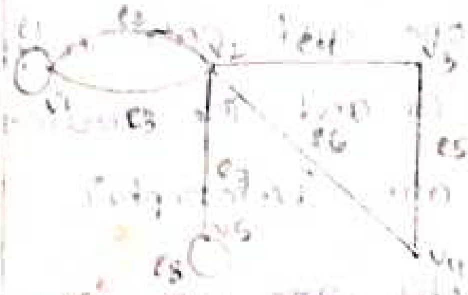
Incidence Matrix :

Let $G = (V, E)$ be an undirected graph. Suppose $v_1, v_2, \dots v_n$ are the vertices and $e_1, e_2 \dots e_m$ are the edges of G. Then the incidence matrix of the given graph is $n \times m$ matrix $M = [m_{ij}]$ where $m_{ij} = \begin{cases} 1 & \text{when edge } e_j \text{ is incident with } v_i \\ 0 & \text{otherwise} \end{cases}$

Incidence matrices can also be used to represent multiple edges and self loops.



$$\begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{array} \begin{array}{c} \begin{array}{cccccc} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 \end{array} \\ \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 \end{bmatrix} \end{array}$$

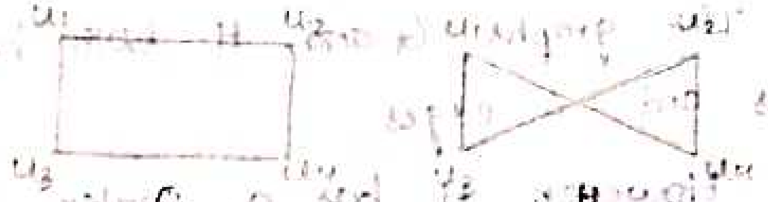


$$\begin{array}{c|cccccccc} & e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & e_7 & e_8 \\ \hline v_1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ v_2 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 \\ v_3 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ v_4 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ v_5 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{array}$$

Isomorphism of Graphs :-

The simple Graph $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ are Isomorphic if there is one-to-one and onto function f from V_2 to V_2 with the property that a and b are adjacent in $G_1 \Leftrightarrow f(a)$ and $f(b)$ are adjacent in G_2, $\forall\ a, b \in V_1$, such a function f is called an Isomorphism and the two graphs G_1 and G_2 are Isomorphic

Eg:- ①



G H

$\deg(u_1) = 2$	$\deg(v_1) = 2$
$\deg(u_2) = 2$	$\deg(v_2) = 2$
$\deg(u_3) = 2$	$\deg(v_3) = 2$
$\deg(u_4) = 2$	$\deg(v_4) = 2$

The following is the Mapping from the vertices of G to the vertices of H

$f(u_1) = v_1$

$f(u_2) = v_3$

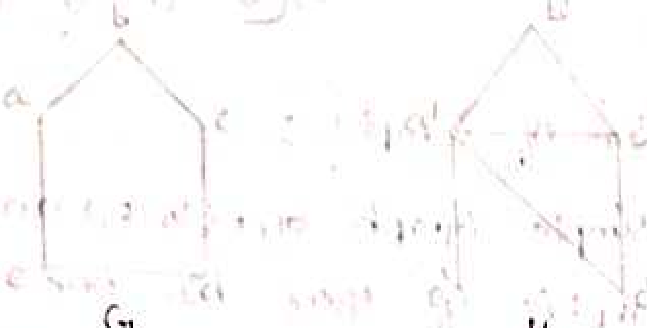
$f(u_3) = v_4$

$f(u_4) = v_2$

Since their exist one-one and onto mapping between the vertices of G and the vertices of H

Therefore G and H are Isomorphic

② Show the following graphs are not isomorphic



G H

$deg(a) = 2$
$deg(b) = 2$
$deg(c) = 3$
$deg(d) = 2$
$deg(e) = 3$

$deg(a') = 4$
$deg(b') = 2$
$deg(c') = 3$
$deg(d') = 2$
$deg(e') = 1$

The graphs G and H both have five vertices and 6 edges

However H has a vertex of degree 1, namely e' and a vertex of degree 4, namely a', where G has a vertices of degree 1 and 4 respectively. It follows that G and H are not isomorphic

<u>Path</u> :

Informally a path is a sequence of edges that begins at a vertex of a graph and travels along the edges of the graph, always connecting pair of adjacent vertices.

let n be a non negative integer. G is an undirected graph. Then a path of length n from u to v in G is a sequence of n edges $e_1, e_2, \ldots e_n$ of G such that $f(e_1) = \{x_0, x_1\}$, $f(e_2) = \{x_1, x_2\} \ldots f(e_n) = \{x_{n-1}, x_n\}$ where $x_0 = u$ and $x_n = v$

When the graph is simple we denote this path by its vertex sequence i.e., $x_0, x_1 \ldots x_n$. The path is a circuit if it begins and ends at the same vertex. A path or circuit is simple if it doesnot contain the same edge more than once. It is also same for directed graphs.

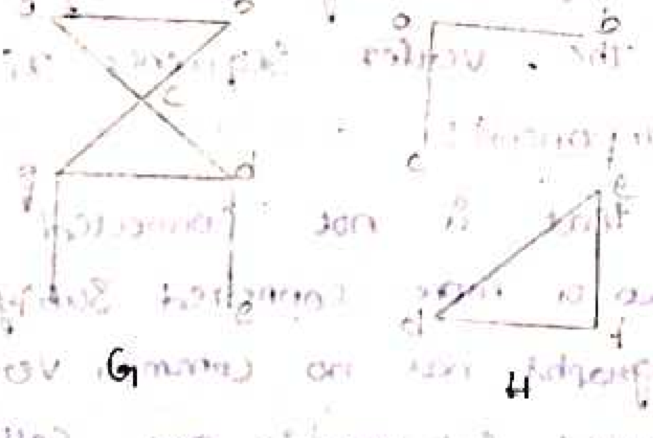
Eg:

a-b-c-d is a Simple Graph

Connectedness of an undirected graph :

An undirected graph is connected if there is a path between every pair of distinct vertices of the graph.

Eg:



The graph G is connected since there is a path between every pair of distinct vertices. However the graph H is not connected for instance there is no path between a and d.

Theorem :

Statement :

There is a simple path between every pair of distinct vertices of a connected undirected graph.

Proof :

Let u and v be two distinct vertices of the connected undirected graph. Since it is connected there is atleast one path between u and v.

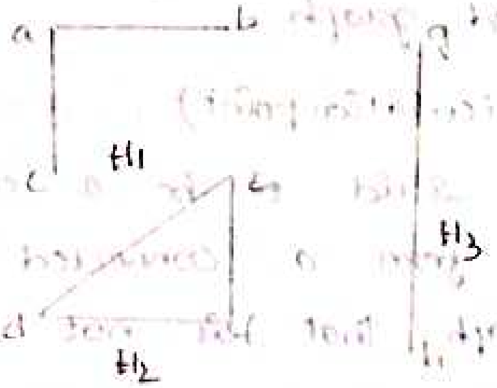
Let $x_0, x_1, \ldots x_n$ where $x_0 = u$, $x_n = v$ be the vertex sequence of a path of least length. This path of least length is simple to see this. Suppose it is not simple. Then $x_i = x_j$ for some i, j with $0 \le i < j \le n$ this means that there is a path from u to v of shorter length with vertex sequence $x_0, x_1, \ldots x_{i-1}, x_j \ldots x_n$ obtained by deleting the edges corresponding to the vertex sequence $x_i \ldots x_{j-1}$

Connected Components :

A graph that is not connected is the union of two or more connected subgraphs, each pair of subgraphs has no common vertex. These disjoint connected subgraphs are called the

Connected component of the graph.

Eg:



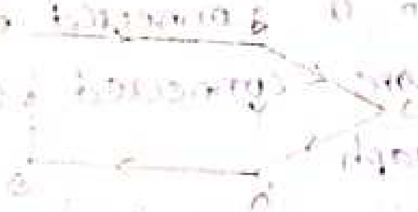
Here H_1, H_2 and H_3 are connected components of G_1.

Connectedness in directed graph :

1. Strongly connected :

A directed graph is strongly connected if there is a path from a to b and b to a whenever a and b are vertices in the graph

Eg:



G_1

Here G_1 is strongly connected graph because any two vertices have a path.

2. Weakly Connected :

A directed graph is weakly connected if there is a path between every two vertices in the underlined undirected graph.

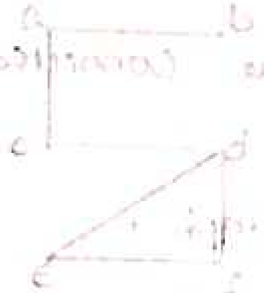
Eg:

The above graph is weakly connected since there is no directed path from a to b but there is a path between any pair of vertices in the underlined graph.

<u>Cut vertex</u> : (Articulation point)

A vertex is said to be a cut vertex if it is removed from a connected graph produces a subgraph that is not connected

<u>Eg</u>:

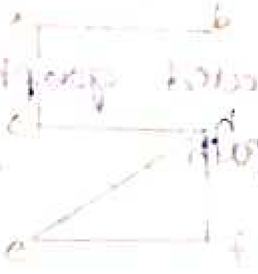


In the above graph a, c, d are the cut vertices

<u>Cut edge</u> (<u>Bridge</u>) :

An edge is said to be a cut edge if it is removed from a connected graph produces a graph with more connected components then in the original graph.

<u>Eg</u>:



In the above graph {c, d} is a cut edge

* <u>Theorem</u>:

<u>Statement</u>:

let G be a graph with adjacency matrix with respect to the ordinary $v_1, v_2, \ldots v_n$ the number of different paths of length r from v_i to v_j where r is a possible integer equals the $(i,j)^{th}$ Entry of A^r

Proof:

The theorem will be proved used mathematical index. let G be a graph with a adjacency matrix A the no. of paths from v_i to v_j of length '1' is the i^{th} energy of m, since this entry is the no. of edges from v_i to v_j. Assume that $(i,j)^{th}$ entry of A^r is the no. of different paths of length r from v_i to v_j. This is induction

$$\text{Since } A^{r+1} = A^r \cdot A$$

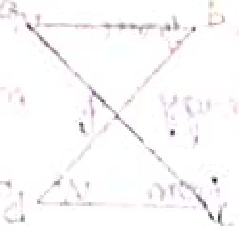
The i^{th} entry of A^{r+1} equals $b_{i1} a_{1j} + b_{i2} + a_{2j} + \dots + b_{in} a_{nj}$, where b_{ij} is the ik^{th} entry of A^r By the induction hypothesis b_{ik} is the no. of paths of length are from v_i to v_k

$$A^r = \begin{bmatrix} b_{11} & b_{12} & \dots & b_{1n} \\ b_{21} & b_{22} & \dots & b_{2n} \\ \vdots & & & \\ b_{i1} & b_{i2} & \dots & b_{in} \\ \vdots & & & \\ b_{n1} & b_{n2} & \dots & b_{nn} \end{bmatrix} \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & & & \\ a_{i1} & a_{i2} & \dots & a_{in} \\ \vdots & & & \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix}$$

A path of length $r+1$ from v_i to v_j is made up of length r from v_i to some intermediate vertex v_k and an edge from v_k to v_j by the product rule for counting the no. of such paths is product of the no. of paths of length r from v_i to v_k namely v_k and the no. of edges from v_k to v_j namely. A_{kj} when the products are added for all positive intermediate vertices v_{kj}, the deside result follows by the sum rule for counting hence the theorem

* Program

How many paths of length four are there from a to d in the simple given below.



∴ The adjacency matrix of given graph

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

Hence the no. of paths of length 4 from a to d is $(1,4)^{th}$ entry of A^4. Now we calculate

$$A^2 = A \cdot A$$

$$A^2 = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}$$

$$= \begin{bmatrix} 0+1+1+0 & 0+0+0+0 & 0+0+0+0 & 0+1+1+0 \\ 0+0+0+0 & 1+0+0+1 & 1+0+0+1 & 0+0+0+0 \\ 0+0+0+0 & 1+0+0+1 & 1+0+0+1 & 0+0+0+0 \\ 0+1+1+0 & 0+0+0+0 & 0+0+0+0 & 0+1+1+0 \end{bmatrix}$$

$$= \begin{bmatrix} 2 & 0 & 0 & 2 \\ 0 & 2 & 2 & 0 \\ 0 & 2 & 2 & 0 \\ 2 & 0 & 0 & 2 \end{bmatrix}$$

$$A^3 = \begin{bmatrix} 2 & 0 & 0 & 2 \\ 0 & 2 & 2 & 0 \\ 0 & 2 & 2 & 0 \\ 2 & 0 & 0 & 2 \end{bmatrix} \cdot \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}$$

$$= \begin{bmatrix} 0+0+0+0 & 2+0+0+2 & 2+0+0+2 & 0+0+0+0 \\ 0+2+2+0 & 0+0+0+0 & 0+0+0+0 & 0+2+2+0 \\ 0+2+2+0 & 0+0+0+0 & 0+0+0+0 & 0+2+2+0 \\ 0+0+0+0 & 2+0+0+2 & 2+0+0+2 & 0+0+0+0 \end{bmatrix}$$

$$= \begin{bmatrix} 0 & 4 & 4 & 0 \\ 4 & 0 & 0 & 4 \\ 4 & 0 & 0 & 4 \\ 0 & 4 & 4 & 0 \end{bmatrix}$$

$$A^4 = \begin{bmatrix} 8 & 0 & 0 & 8 \\ 0 & 8 & 8 & 0 \\ 0 & 8 & 8 & 0 \\ 8 & 0 & 0 & 8 \end{bmatrix}$$

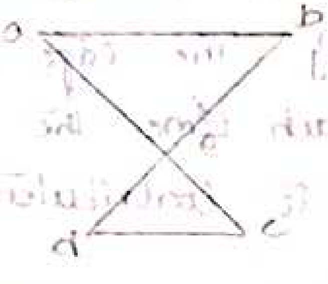
∴ There are exactly 8 paths of length 4 from a to d the paths are :

1) a-b-a-c-d
2) a-b-a-b-d
3) a-c-a-b-d
4) a-c-a-c-d
5) a-b-d-b-d
6) a-b-d-c-d
7) a-c-d-c-d
8) a-c-d-b-d

Eular circuit and Eular path :

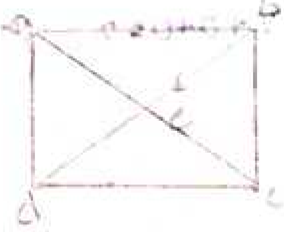
An Eular circuit in a graph G is a simple circuit containing every edge of G. An eular path in G is a simple path containing every edge of G.

Eg:



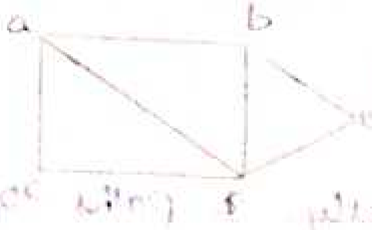
This graph is having a eular circuit a-e-c-d-e-b-a

Eg:



a-b-e-d-c-e-a-d

This graph has neither eular circuit nor eular path.



This graph having a eular path namely a-b-e-d-c-a-d-b but it doesn't have a eular circuit.

Theorem:

A connected Multigraph has an eular circuit $\Leftrightarrow$ each of its vertices has even degree

(or)

Necessary and sufficient condition for eular circuit

Proof:

Suppose G is a connected multigraph and it has an eular circuit

We have to show that every vertex must have even degree

First note that an eular circuit begins with a vertex 'a' and continues with an edge incident on 'b' say {a,b}. The edge {a,b} contributes one to deg(a). Each time the circuit passes through a vertex it contributes two to the vertex degree finally the circuit terminates where it is started, contributing one to deg(a)

Therefore deg(a) must be even because the circuit contributes one when it begins, one when it terminates two every time it passes through 'a'. A vertex other than 'a' has even degree because the circuit contributes to its degree, each time it passes through the vertex. Hence we can conclude that if a connected graph as an either euler circuit then every vertex must have even degree.

Conversely suppose that G is a connected Multigraph and the degree of every vertex of G is even

We have to show that G has an Euler circuit

We will have a simple circuit that begins at an arbitary vertex 'a' of G

let $x_0 = a$, first we arbitarly whose an edge $\{x_0, x_1\}$ incident with 'a', we continue by building a simple path $\{x_0, x_1\}, \{x_1, x_2\} \ldots \{x_{n-1}, x_n\}$ as long as possible.

Since G is connected and it has finite number of edges the path terminates. It begins at a vertex 'a' with an edge of the form $\{a, x\}$ and terminates at 'a' with an edge of the form $\{y, a\}$. This follows because each time the path goes through with even degree it was uses only one edge to entre the vertex. So there atleast one edge remain for the path to leave the vertex

This path may use all the edges (or) may not

An Euler circuit has been constructed if all edges have been used otherwise consider the subgraph H obtained by from G by deleting the edges already used and vertices that are not incident with any remaining edges. Since G is connected, H has atleast one vertex in common with the circuit that has be deleted. Let 'w' be such a vertex. Every vertex in H has even degree. Begin at w construct a simple path in H by choosing edges as long as possible. This path must terminate at 'w' Next from a circuit in G by combining the circuit H with the original circuit in G. continue this process untill all edges have been included in the circuit. The construction shows that if the vertices of a connected multigraph has all the vertices with even degree. Then the graph as an Euler circuit. Hence, the theorem

Theorem:

Statement:

A connected Multigraph has an Euler path but not an Euler circuit $\Leftrightarrow$ it has exactly two vertices of odd degree

Proof:

First suppose that a connected multigraph as Euler path from a to b but not an Euler circuit.

The first edge of the path contributes one to the (degree of 'a') deg(a). A contribution of two to the deg(a) is made every time the path passes to 'a'.

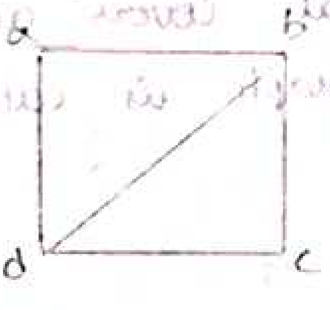
The least edge in the path contributes one to the deg(b). Every time the path passes to 'b' there is a contribution two to its degree.

Consequently both 'a' and 'b' have odd degree. Every other vertex has an even degree. Since the path contributes two to the degree of the vertex whenever it passes through it.

Now, consider the converse suppose G is a connected multigraph and it has exactly two vertices of odd degree say a,b consider the larger graph made up off the original graph with an edge {a,b}, every vertex of this larger graph has even degree. So that there exists an Euler circuit. The removely of new edge produces the Euler path in the original graph. Hence the theorem.

Eg:

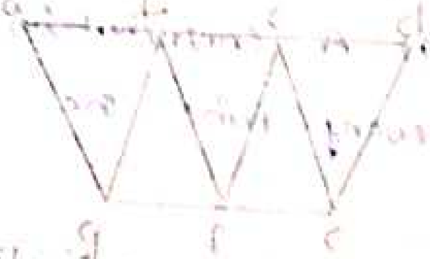
(a)



b-c-d-a-b-d

The above graph contains the Euler path. Since exactly two vertices 'b' and 'd' are having odd degree and the remaining vertices are of even degree.

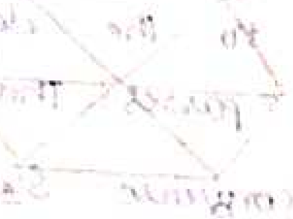
(b)



g - a - b - g - f - b - c - f - e - c - d - e

The above graph contains an Euler path since there are exactly two vertices 'e' and 'g' of odd degree and the remaining vertices are of even degree.

(c)



The above graph has 6 vertices of odd degree. So, it will not have an Euler path.

<u>Hamilton</u> <u>paths</u> & <u>circuits</u> :-

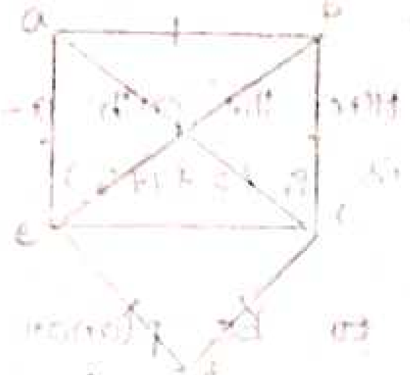
A path $x_0, x_1, \ldots, x_n$ in the Graph $G=(V,E)$ is called Hamilton path if $V = \{x_0, x_1, \ldots, x_n\}$ and $x_i \neq x_j$ for $0 \le i < j \le n$.

(or)

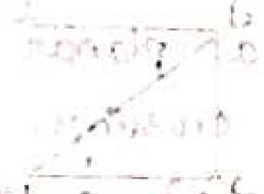
A simple path that covers all the vertices in the given graph is called Hamilton path.

A circuit $x_0, x_1, \ldots, x_n, x_0$ with $n > 1$ in a graph G is called Hamilton circuit. If $x_0, x_1, \ldots, x_n$, is a Hamilton path

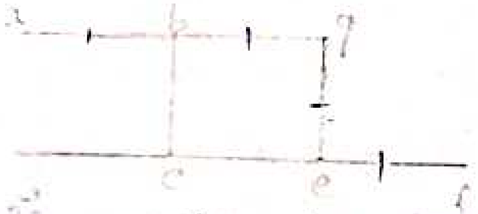
Eg:



In this graph there exists a Hamilton circuit i.e., a, b, c, d, e, a



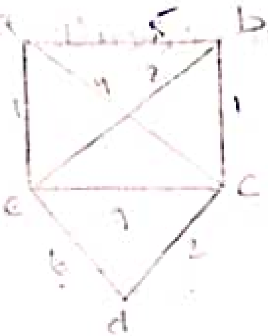
In this above graph no Hamilton circuit, but a Hamilton path a, b, c, d

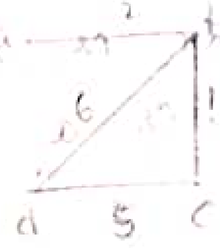


In the above graph there is neither a Hamilton circuit nor Hamilton path. Since any path containing all vertices must contain one of the edges {a, b}, {e, f} and {c, d} more than once

Weighted Graph :-

Graphs that have a number assigned to each edge are called weighted graph. The length of the path in a weighted graph be the sum of the weights of the edges in the path.





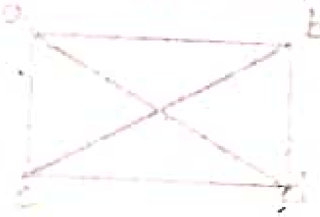
Here the path a-b-c-d length is 8 (2+1+5)

Planar Graph :-

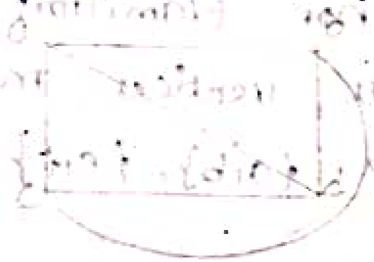
A Graph is said to be planar if it can drawn in the plane without any edge crossings. Such a drawing is called a planar representation of the graph.

A Graph may be a planar even if it is usually drawn with crossings, since it may be possible to draw it in a different way without crossings.

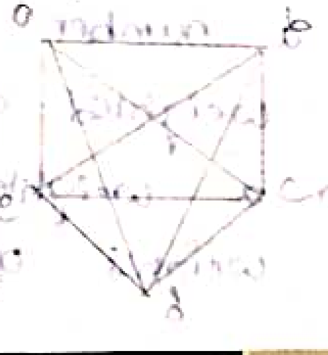
Eg: 1.



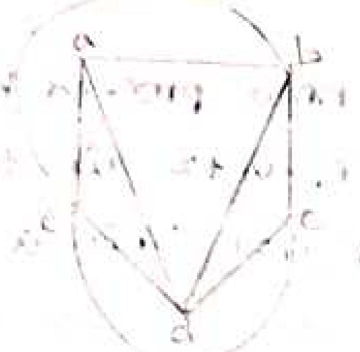
The above graph is planar, since it has a following planar representation



2.



It is not planar since we cannot drawn an edge a-c without edge crossing in the planar representation of the graph.



Euler's Formula :-

Let G be a connected planar simple graph with e edges and v vertices. Let r be the number of regions in the planar representation of G. Then $r = e - v + 2$

Proof:

First we specify a planar representation of G.

We will prove the theorem by constructing a sequence of subgraphs $G_1, G_2 \ldots G_e = G$.

Successively adding an edge at the each stage. This can be done using the following inductive definition. Orbitarily pick one edge of G to obtain G_1, Obtain G_n from G_{n-1} by orbitarily adding an edge that is, incident with a vertex already in G_{n-1}. Adding the other vertex incident with this edge if it is not already in G_{n-1}. This construction is possible since G is connected. It can be obtained after 'e' edges are added. Let r_n, e_n and v_n represent the number of regions, the edges and vertices of the planar representation of G_n

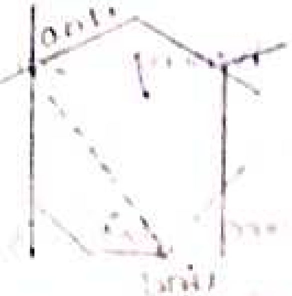
The proof will now proceed by induction the relationship $r_1 = e_1 - v_1 + 2$ is true for G_1.. Since $e_1 = 1, v_1 = 2$ and $r_1 = 1$. This is shown in the following figure.

Now, we assume that $r_n = e_n - v_n + 2$. This is the induction hypothesis.

Induction step ;

Let $\{a_{n+1}, b_{n+1}\}$ be the edge that is added to G_n to obtain G_{n+1}. There are two possibilities to consider. In the first case both a_{n+1} and b_{n+1} are already in G_n. These two vertices must be on the boundary of the common region or else it would be impossible to add the edge $\{a_{n+1}, b_{n+1}\}$ to G_n without two edges crossing. In this case $r_{n+1} = r_n + 1$, $e_{n+1} = e_n + 1$, $v_{n+1} = v_n$

Thus each side of the formula increase by exactly one. So the formula is true in other words $r_{n+1} = e_{n+1} - v_{n+1} + 2$. This is illustrated from the following figure.



In the second case one of the two vertices of the new edge is not already in G_n. Suppose a_{n+1} is in G_n but b_{n+1} not adding this new edge doesnot produce any new regions, since b_{n+1} must be in a region that as a_{n+1} on its boundary. Then

$$r_{n+1} = r_n$$
$$e_{n+1} = e_n + 1$$
$$v_{n+1} = v_n + 1$$

Each side of the formula increases by exactly '1'. So this formula is true in other-words $r_{n+1} = e_{n+1} - v_{n+1} + 2$. This is illustrated in the following figure

This completes the induction proofs. Hence $r_n = e_n - v_n + 2 \ \forall \ n \geq 1$

1) Suppose that a connected planar simple graph has 20 vertices, each of degree 3 into how many regions does the representation of this planar graph split the plane.

Since the graph having 20 vertices

Then $v = 20$

Each vertex degree is 3 and we know that degree sum of the vertices of a graph is equal to twice the number of edges i.e.,

$$2e = \sum_{v \in V} \deg(v)$$

$$2e = 20 \times 3$$

$$2e = 60$$

$$\therefore e = 30$$

By Euler's formula, the number of regions in the planar representation of graph

$$r = e - v + 2$$

$$r = 30 - 20 + 2$$

$$\therefore r = 12$$

Theorem:

If G is a connected planar graph with the planar 'e' edges and 'v' vertices where $v \geq 3$ then $e \leq 3v - 6$

Proof:

A connected planar graph drawn in the plane divides the plane into 'r' number of regions. The degree of each region is atleast 3. Since, there are atleast 3 vertices in the graph.

The sum of the degrees of the region is exactly twice the number of edges of the graph. Since each region has degree greater than or equal to 3. It follows that

$$2e = \sum_{\text{all regions } r} \deg(r) \geq 3r$$

Hence $2e \geq 3r$

$\Rightarrow \frac{2e}{3} \geq r$ (∵ Euler formula $r = e - v + 2$)

$\Rightarrow \frac{2e}{3} \geq e - v + 2$

$\Rightarrow 2e \geq 3e - 3v + 6$

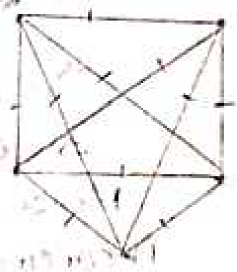
$\Rightarrow 3v - 6 \geq 3e - 2e$

$\therefore e \leq 3v - 6$.

Example:

Show that K_5 is not planar.

Sol The Number of vertices in K_5



$v = 5$ and Edges $e = 10$

$e \leq 3v - 6$

$10 \leq 3(5) - 6$

$10 \leq 15 - 6$

$10 \leq 9$

$\therefore e = 10 \nleq 9$ [∵ $e \leq 3v - 6$]

Since it not satisfying the formula so it is Non-planar

Theorem :-

If a connected planar simple graph has e edges and v vertices with $v \geq 3$ and no circuites of length 3, then $e \leq 2v - 4$.

Proof:

A connected planar simple graph divides the plane into 'r' number of regions

Since each degree regions degree is atleast four(4) and we also know that the degree sum of the regions is equal to twice the number of edges.

i.e. $2e = \sum_{\text{all } R \text{ Regions}} deg(R) \geq 4r$

$\Rightarrow 2e \geq 4r$

$\Rightarrow e \geq 2r$

$\Rightarrow e/2 \geq r$

$\Rightarrow e/2 \geq e - v + 2$ (∵ Euler's formula)

$\Rightarrow e \geq 2e - 2v + 4$

$\Rightarrow 2v - 4 \geq 2e - e$

$\Rightarrow 2v - 4 \geq e$

$\Rightarrow e \leq 2v - 4$

Hence, the Theorem.

Theorem :-

If G is a connected planar simple graph then G has a vertex of degree not exceeding 5

Proof :

If G has one or two vertices the result is true.

If G has atleast 3 vertices we know that

$$e \leq 3v - 6$$

$$2e \leq 6v - 12 \quad \text{--- (1)}$$

If the degree of every vertex is atleast 6 then By Hand shaking theorem

$$2e = \sum_{v \in V} \deg(v)$$

(2)

Then we would have $2e \geq 6v$. But this contradicts the above inequality.

It follows that there must be a vertex with degree not exceeding 5.

Elementary Subdivision :-

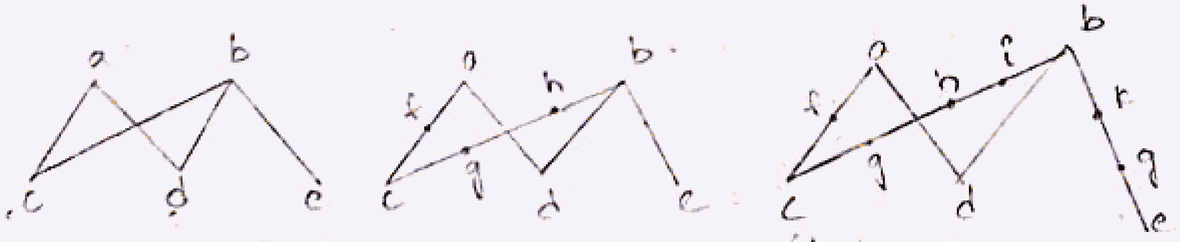
If a graph is planar then any graph obtained by removing an edge $\{u, v\}$ and adding a New vertex w together with edges $\{u, w\}$ and $\{w, v\}$. Such an operation is called elementary subdivision.

Homomorphism :-

$G_1 = \langle V_1, E_1 \rangle$, $G_2 = \langle V_2, E_2 \rangle$ are called Homomorphism. If they can be obtained from the same graph by a sequence of elementary sub-divisions.

Eg :

The Following three graphs are Homomorphic since all can be obtained from the first graph by sequence of elementary subdivisions



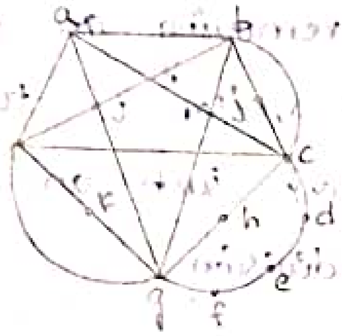
(G_1) (G_2) (G_3)

Kuratowski's Theorem :-

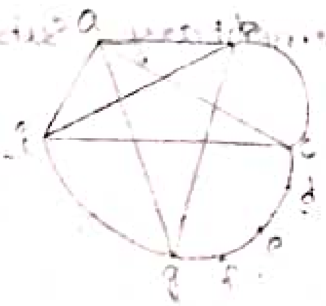
A graph is non planar iff and only iff it contains a subgraph homomorphic to K_{33} or K_5.

Example :-

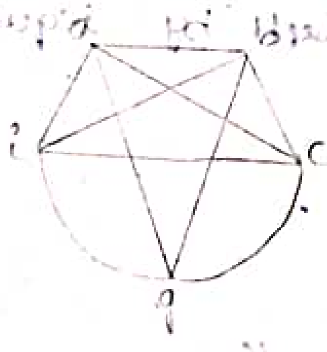
Determine whether the graph G shown in the following figure is planar.



G has a subgraph H homomorphic to K_5. H is obtained by deleting h, j and k and all edges incident with these vertices. H is homomorphic to K_5, since it can be obtained from K_5 by a sequence of elementary subdivisions i.e., adding the vertices d, e, f. Hence G is not planar.



(H).



(K_5)

Graph Coloring :-

A Coloring of a Simple graph is the assignment of the color to each vertex of the graph such that no two adjacent vertices are assigned the same color

The chromatic number of a graph is the least number of colors required for coloring of a graph It is denoted by $\chi(G)$

Eg: 1

The above graph chromatic number $\chi(G) = 4$

Eg: 2

What is the chromatic number of K_n

Coloring of K_n can be constructed using n colors by assigning a different color to each vertex, since no two vertices can be assigned the same color because every two vertices of the graph are adjacent. Hence the chromatic number of K_n = n

Eg: 3

What is the chromatic number of graph C_n

Two colors are needed to color C_n when n is even, three are needed when n is odd

Theorem :-

A connected simple graph is bi-partite graph if and only if it is two colorable (chromatic number = 2)

Proof :

let G be a connected simple graph and suppose it is a bi-partite graph. Then by its definition its vertex set is divided into two sets such that no two vertices in the same set are adjacent. Let the vertex sets are V_1 and V_2.

Since all the vertices in V_1 are not-adjacent, then we can give color C_1 to all of them. Similarly we can give color C_2 to all the vertices of V_2

Hence we can color the graph using two colors therefore the chromatic number of G is 2.

Conversely, suppose that G is two colorable. We can divide the vertex set of G into two sets V_1 and V_2 based on their color. Let V_1 contain the vertices which are having color C_1. Similarly V_2 contains the vertices which are having color C_2. Since no two vertices in V_1 or in V_2 are adjacent and the edges are in between the vertices of V_1 and the vertices of V_2. Therefore G is a bi-partite graph hence the theorem.

Four color theorem :-

Initially people believe that the chromatic number of a planar graph is no greater than 4 i.e., any planar graph can be colored by using less than or equal to 4 colors. After that it is proved that it is not always positive i.e., the theorem proved is wrong.

Travelling Salesman problem :-

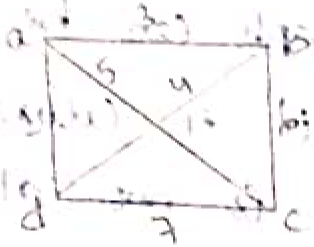
A travelling salesman has visit each of cities exactly ones and return to his starting point i.e., the problem asks for the circuit of minimum total weight in a waited complete undirected graph, that visits each vertex exactly ones and return to his starting point. i.e., asking for a Hamiltonian circuit with minimum total way in the complete graph.

The most straight forward way to solve an instance of the travelling salesman problem is to examine all possible hamilton circuits and select one of the minimum total length. ones the starting point is choosen then there are $(n-1)!$ different Hamilton circuits to examine.

Since a Hamilton circuit can be traversed in reverse order we need only to examine $\frac{(n-1)!}{2}$ circuits to find out the answer.

Example:

Solve a travelling salesman problem given below



1) $a \xrightarrow{3} b \xrightarrow{6} c \xrightarrow{7} d \xrightarrow{2} a$ of length 18

2) $a \xrightarrow{5} c \xrightarrow{7} d \xrightarrow{4} b \xrightarrow{3} a$ of length 19

3) $a \xrightarrow{2} d \xrightarrow{4} b \xrightarrow{6} c \xrightarrow{5} a$ of length 17

The solution is $a-d-b-c-a$ of length 17

Dijkstra's Algorithm :-

Algorithm Dijkstra's (G : weighted connected simple graph with +ve weights)

{ G has vertices $a = v_0, v_1, \dots v_n = z$ and weights $w(v_i, v_j) = \infty$ if $\{v_i, v_j\}$ is not an edge in G }

for $i := 1$ to n

$L(v_i) := \infty$

$L(a) := 0$

$S := \phi$

while $z \notin S$

begin

$u :=$ a vertex not in S with $L(u)$ is minimum

$S := S \cup \{u\}$

for all vertices v not in S

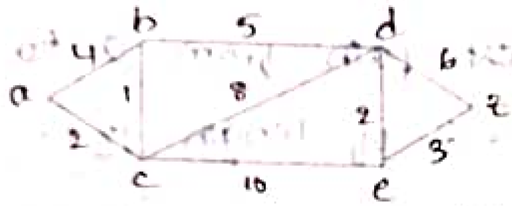
if $L(u) + w(u,v) < L(v)$ then $L(v) := L(u) + w(u,v)$

end

$\{L(z)$ = length of the shortest path from a to $z\}$

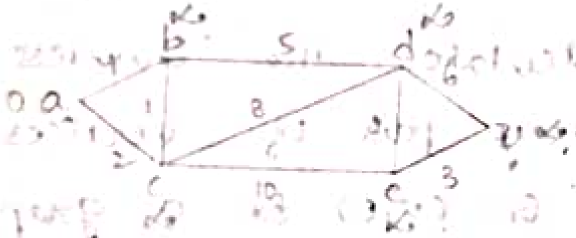
The above algorithm is also known as "Single Source Shortest path algorithm".

Eg: Find the shortest path from a to z

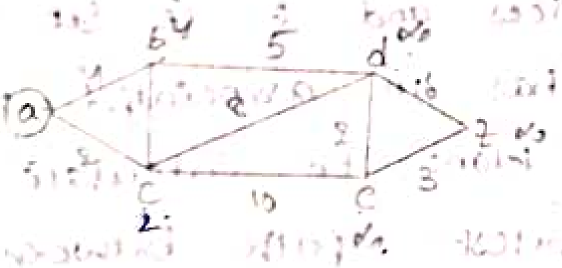


Sol

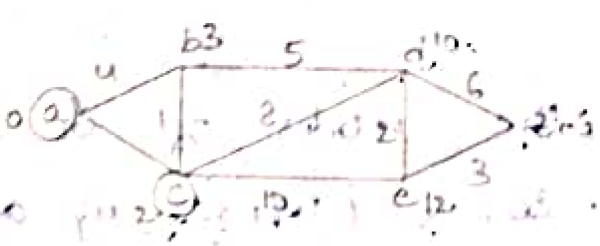
Step-1:



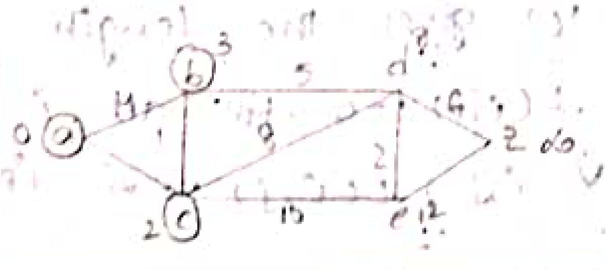
Step-2:



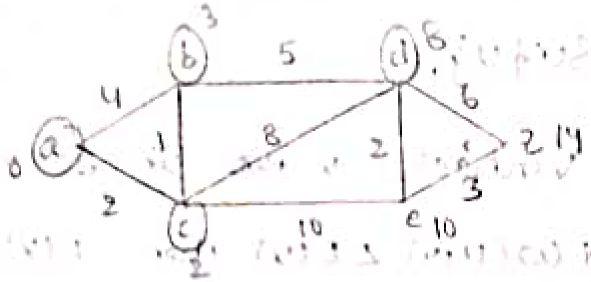
Step-3:



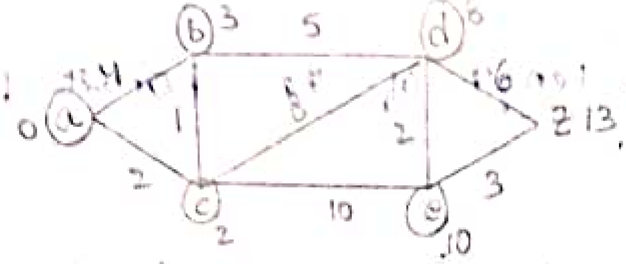
Step-4:



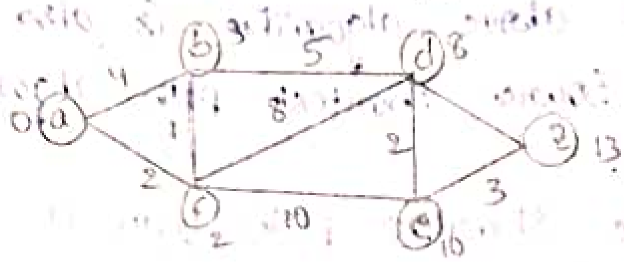
Step-5:



Step-6:



Step-7:



The Shortest path from a to z is a-c-b-d-e-z of length 13.

Floyd's Algorithm :-

This calculates the shortest path length between every pair of vertices in the graph. Let $G = (V, E)$ be a graph, V be the set of vertices and E is set of Edges, Each edge has an associated non-negative length. We want to calculate the length of the shortest path between each pair of vertices.

Suppose, the nodes of G are numbered from 1 to n. So, $V = \{1, 2, 3 \dots n\}$ and suppose a matrix 'L' gives the length of each edge with $L(i,i) := 0$ for all $i = 1, 2, \dots n$, $L(i,j) \geq 0 \ \forall \ i,j$, $L(i,j) := \infty$ if the edge {i,j} does not exist

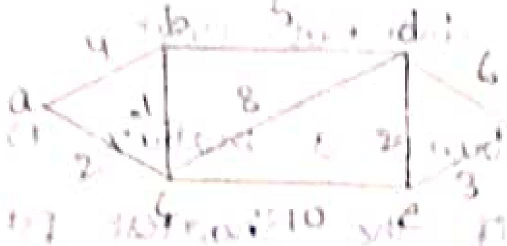
We construct a matrix D that gives the length of the shortest path between each pair of nodes. The algorithm initializes 'D to L'. Then it does iteration after kth iteration D gives the length of the shortest path that only used nodes in set of {1,2,... k} as an intermediate nodes. After n iterations D gives the length of the shortest path using any of the n nodes in V as an intermediate node, which is the result we want. If D_k represents the matrix D after k^{th} iteration, the necessary check can be implemented by $D_k[i,j] \leftarrow$

$$\min\{D_{k-1}[i,j],\ D_{k-1}[i,k] + D_{k-1}[k,j]\}$$

length

<u>Algorithm</u>:

```
Algorithm Floyd (L[1,...n; 1,....n])
{
  array D[1...n, 1....n]
  D <- L
  For k <- 1 to n do
  For i <- 1 to n do
  For j <- 1 to n do
  D[i,j] <- min(D[i,j], D[i,k] + D[k,j])
}
```



Sol:

$$\begin{array}{c|c} & \begin{matrix} a & b & c & d & e & f \end{matrix} \\ \begin{matrix} a \\ b \\ c \\ d \\ e \\ f \end{matrix} & \begin{bmatrix} 0 & 4 & 2 & \infty & \infty & \infty \\ 4 & 0 & 1 & 5 & \infty & \infty \\ 2 & 1 & 0 & 8 & 10 & \infty \\ \infty & 5 & 8 & 0 & 2 & 6 \\ \infty & \infty & 10 & 2 & 0 & 3 \\ \infty & \infty & \infty & 6 & 3 & 0 \end{bmatrix} \end{array} = D_0 = L$$

$$D_1 = \begin{array}{c|c} & \begin{matrix} a & b & c & d & e & f \end{matrix} \\ \begin{matrix} a \\ b \\ c \\ d \\ e \\ f \end{matrix} & \begin{bmatrix} 0 & 4 & 2 & \infty & \infty & \infty \\ 4 & 0 & 1 & 5 & \infty & \infty \\ 2 & 1 & 0 & 8 & 10 & \infty \\ \infty & 5 & 8 & 0 & 2 & 6 \\ \infty & \infty & 10 & 2 & 0 & 3 \\ \infty & \infty & \infty & 6 & 3 & 0 \end{bmatrix} \end{array}$$

$$D_2 = \begin{array}{c|c} & \begin{matrix} a & b & c & d & e & f \end{matrix} \\ \begin{matrix} a \\ b \\ c \\ d \\ e \\ f \end{matrix} & \begin{bmatrix} 0 & 4 & 2 & 9 & \infty & \infty \\ 4 & 0 & 1 & 5 & \infty & \infty \\ 2 & 1 & 0 & 6 & 10 & \infty \\ 9 & 5 & 6 & 0 & 2 & 6 \\ \infty & \infty & 10 & 2 & 0 & 3 \\ \infty & \infty & \infty & 6 & 3 & 0 \end{bmatrix} \end{array}$$

$$D_3 = \begin{array}{c} \\ a \\ b \\ c \\ d \\ e \\ f \end{array} \begin{array}{c} \begin{array}{cccccc} a & b & c & d & e & f \end{array} \\ \begin{bmatrix} 0 & 3 & 2 & 8 & 12 & \infty \\ 3 & 0 & 1 & 5 & 11 & \infty \\ 2 & 1 & 0 & 6 & 10 & \infty \\ 8 & 5 & 6 & 0 & 2 & 6 \\ 12 & 11 & 10 & 2 & 0 & 3 \\ \infty & \infty & \infty & 6 & 3 & 0 \end{bmatrix} \end{array}$$

$$D_4 = \begin{array}{c} \\ a \\ b \\ c \\ d \\ e \\ f \end{array} \begin{array}{c} \begin{array}{cccccc} a & b & c & d & e & f \end{array} \\ \begin{bmatrix} 0 & 3 & 2 & 8 & 10 & 14 \\ 3 & 0 & 1 & 5 & 7 & 11 \\ 2 & 1 & 0 & 6 & 8 & 12 \\ 8 & 5 & 6 & 0 & 2 & 6 \\ 10 & 7 & 8 & 2 & 0 & 3 \\ 14 & 11 & 12 & 6 & 3 & 0 \end{bmatrix} \end{array}$$

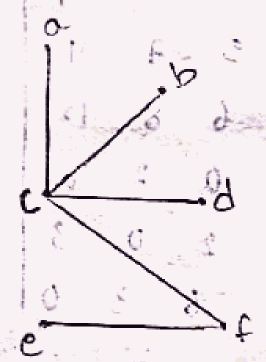
$$D_5 = \begin{array}{c} \\ a \\ b \\ c \\ d \\ e \\ f \end{array} \begin{array}{c} \begin{array}{cccccc} a & b & c & d & e & f \end{array} \\ \begin{bmatrix} 0 & 3 & 2 & 8 & 10 & 13 \\ 3 & 0 & 1 & 5 & 7 & 10 \\ 2 & 1 & 0 & 6 & 8 & 11 \\ 8 & 5 & 6 & 0 & 2 & 5 \\ 10 & 7 & 8 & 2 & 0 & 3 \\ 13 & 10 & 11 & 5 & 3 & 0 \end{bmatrix} \end{array}$$

Trees

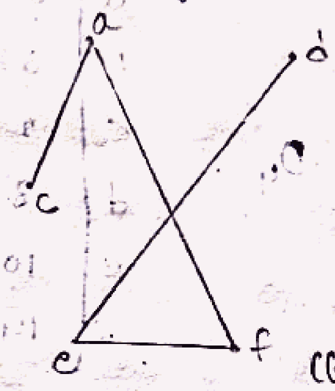
Tree:-

A tree is a connected undirected graph with no simple circuits. Since a tree cannot have a simple circuit it cannot contain multiple edges or loops. Therefore any tree must be a simple graph

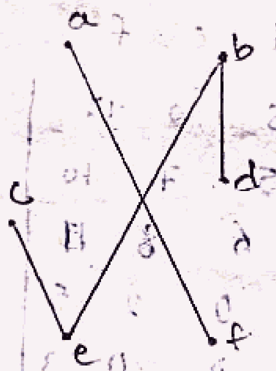
Eg:



(G_1)



(G_2)



(G_3)

Here G_1 and G_2 are trees since they are connected acyclic. G_3 is not a tree since it is not connected, G_3 is a forest

Note:

Graphs contain no simple circuits which are not connected can be called as Forests. They have the property that each of their connected components is a tree.

Theorem :-

An undirected graph is a tree iff only iff there is a unique simple graphpath between any two of its vertices

Proof :-

First Assume that T is a tree.
Then it is a connected graph with no simple circuits.
Let x and y be two vertices of V. Since T is connected, there is a simple path between x and y.
Moreover this path must be unique, if there was a second search path, the path formed by combining the first path from x to and y followed by the path from y to x would form a circuit.
It is a contraduction.
Hence there is a unique simple path between any two vertices of a tree.
Conversely, suppose that there is a unique simple path between any two vertices of a graph T. Then T is connected

Further more T has no simple circuits
To see that suppose T had a simple circuit that contain the vertices x and y
Then there would be two simple paths between x and y and a second simple

path from y to z. This is a contradiction.

Hence, a graph with unique simple path between any two vertices is a tree.

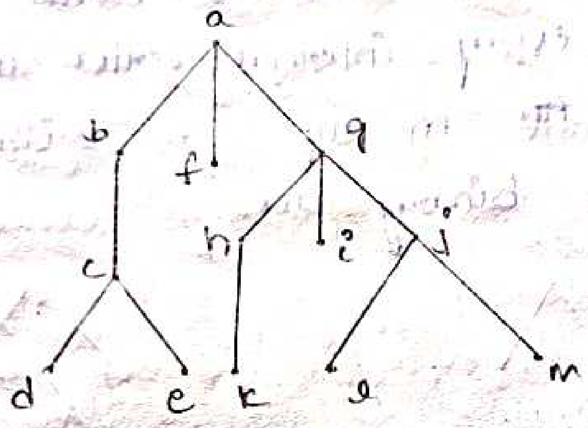
Hence the theorem.

Rooted Tree :-

A Rooted tree is a tree in which one vertex has been designated as the root and every edge is directed away from the root. Suppose T is a rooted tree, if v is a vertex other than the root, the parent v is the unique vertex u such that there is a directed edge from u to v. When u is the parent of v, v is called child of u. Vertices with the same parent are called siblings. The Ansisters of the vertex other than the root are the vertices in the path from root to this vertex, Exiluding the vertex itself and including the root. The descendence of a vertex v are those vertices that have v as an Ansister.

A vertex of a tree is called a leaf if it has no children vertices that have children are called internal vertices. The root is an internal vertex unless it is the only vertex in the graph. In which it is a leaf. If a is a vertex in a tree, the subtree with 'a' as its root is the subgraph of the tree consisting of 'a' and its descendence and all edges incident to these descendence.

Eg:-



In the above rooted tree find the parent of c, the children of G, the siblings of h, all Ancestors of e, all descendence of b, all internal vertices, all leafes and the subtree rooted at g.

(i) parent of 'c' is 'b'

(ii) children of 'g' are h, i, j

(iii) Siblings of h are i and j

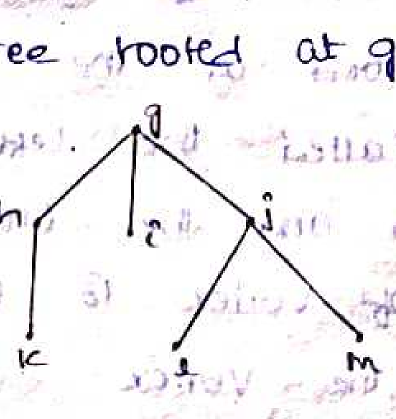
(iv) Ancestors of e are a, b, c

(v) Decendence of b are c,

(vi) Internal ~~vertices~~ Nodes are a, b, c, g, h, j

(vii) The leaves are f, i, k, l, m

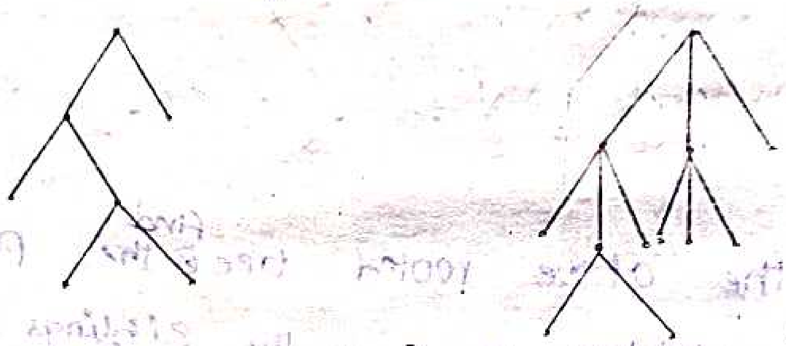
(viii) The subtree rooted at g is



m – ary Tree :–

A rooted tree is called an m-ary tree. If every internal vertex has no more than m children. The tree is called a full m-ary tree. If every internal vertex has exactly m children. An m-ary tree with $m=2$ is called a binary tree.

Eg:



T_1 T_2

T_1 is a fully binary tree

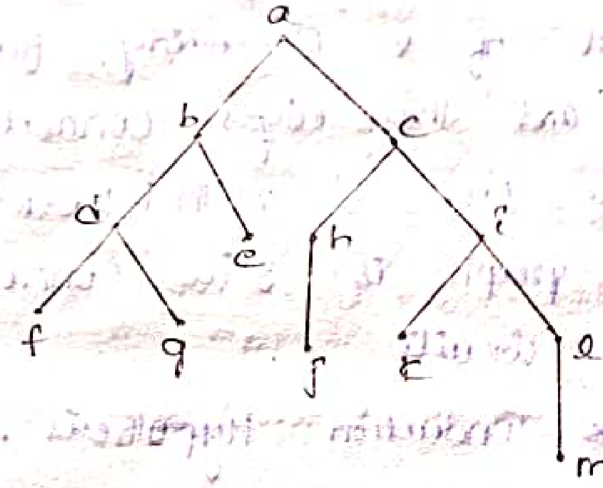
T_2 is a fully 3-ary tree.

Ordered rooted tree :–

An ordered rooted tree is a rooted tree where the children of each internal vertex are ordered. In an ordered binary tree if an internal vertex has two children the first child is called left child and the second child is called the right child. The three rooted at the left child of the vertex is called the left subtree of there vertex, and the tree rooted at the right child of vertex is called the right subtree of the vertex.

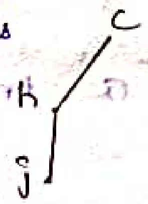
Eg:

What are the left and right children of d in binary tree (T) shown in the figure and what are the left and right structure Subtrees of c

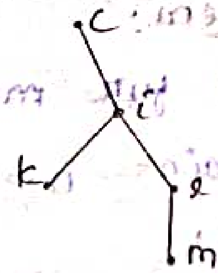


Sol

* The left children of d is 'f'
* The Right children of d is g
* The left Subtree of c is



* The Right Sub tree of c is



Theorem :-

A tree with n vertices has (n-1) edges.

Proof:

We will use Mathematical Induction to prove this theorem.

Suppose n=1, a tree with n=1 vertex has no edges.

$\therefore$ It is tree for $n=1$

Assume that Every tree with k vertices has $k-1$ edges, where k is positive integer. Suppose that a tree T has $k+1$ vertices and v is a leaf of T and left w be the parent of v. Removing from T the vertex v and the edges connectives w to d. Produce a tree t' with k vertex. Since the resulting graph is still connected and has no simple circuits.

By the Induction Hypothesis, T has $k-1$ edges.

It follows that T has k edges since it has one more than T', the edge connecting v and w.

Hence a tree with n vertices has $n-1$ edges.

Theorem:

A full m-ary tree with i Internal vertices contain $n = mi+1$ vertices.

Proof:

Every vertex Except the root is the children of an Internal vertex. Since each of the i internal vertices has 'm' children, there $m \cdot i$ vertices in the tree other than the root.

The tree contains $n = mi+1$ vertices.

Theorem:-

A full m-ary tree with the following

(i) n vertices has $i = (n-1)/m$ internal vertices and $l = [(m-1)n+1]/m$ leaves.

(ii) i Internal vertices has $n = mi+1$ vertices and $l = (m-1)i+1$ leaves.

(iii) l Leaves has $n = \frac{(ml-1)}{(m-1)}$ vertices and $i = \frac{(l-1)}{(m-1)}$ Internal vertices.

Proof:

Let M be the Number of vertices, 'i' be the number of Internal vertices and 'l' be the number of leaves.

We know that, $n = mi+1$ in a full m-ary tree and the Number of vertices $n = i+l$

$$\therefore n = mi+1 \text{ --- } (1)$$

$$n = i+l \text{ --- } (2)$$

For Statement - (1)

Since from - (1) $\Rightarrow n = mi+1$

$$\Rightarrow n-1 = mi$$

$$\Rightarrow i = \frac{(n-1)}{m}$$

from - (2), $n = i+l$

$$\Rightarrow n = \frac{(n-1)}{m} + l$$

$$\Rightarrow n - \frac{(n-1)}{m} = l$$

$$\Rightarrow \frac{nm - n+1}{m} = l$$

$$\Rightarrow \frac{n(m-1)+1}{m} = l$$

$$\therefore l = \frac{[(m-1)n+1]}{m} \text{ --- } (3)$$

For Statement - (2)

from - (1), $n = mi + 1$

from (2), $n = i + l$

by (1) $\Rightarrow mi + 1 = i + l$

$\Rightarrow l = mi - i + 1$

$\therefore l = (m-1)i + 1$ — (4)

For Statement - (3)

from - (3), $l = \frac{(m-1)n + 1}{m}$

$\Rightarrow lm = (m-1)n + 1$

$\Rightarrow lm - 1 = (m-1)n$

$\Rightarrow n = \frac{(lm-1)}{(m-1)}$

from (4), $l = (m-1)i + 1$

$l - 1 = (m-1)i$

$\therefore i = \frac{(l-1)}{(m-1)}$

Level and Height :-

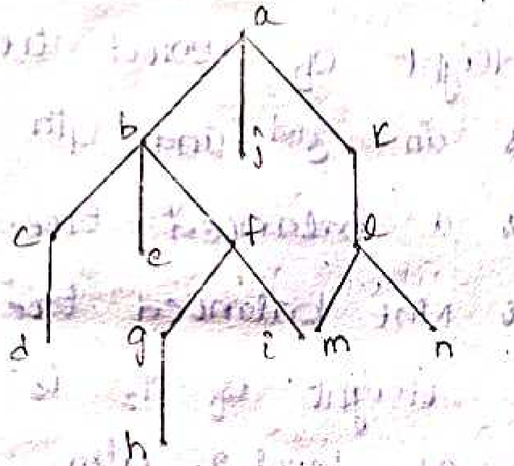
The level of vertex v in a rooted tree is the length of the unique path from root to this vertex.

The level of a root is defined to be zero (0).

The Height of the rooted tree is the maximum of the levels of the vertices. In other words, the height of the rooted tree is the length of the longest path from root to any vertex.

Eg:

Find the level of each vertex in the rooted tree shown below. What is the height of a tree.



Sol:

The level of a $L(a) = 0$ $L(i) = 3$

$L(b) = 1$ $L(j) = 1$

$L(c) = 2$ $L(k) = 1$

$L(d) = 3$ $L(l) = 2$

$L(e) = 2$ $L(m) = 3$

$L(f) = 2$ $L(n) = 3$

$L(g) = 3$

$L(h) = 4$

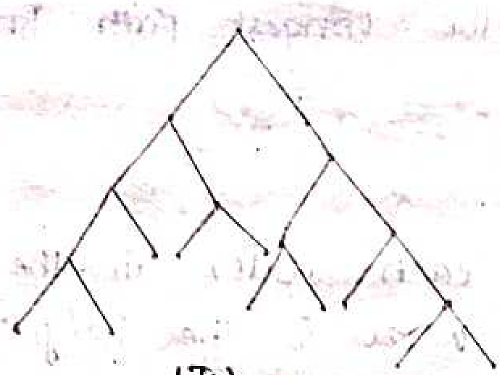
$\therefore$ The Height of the Tree is 4

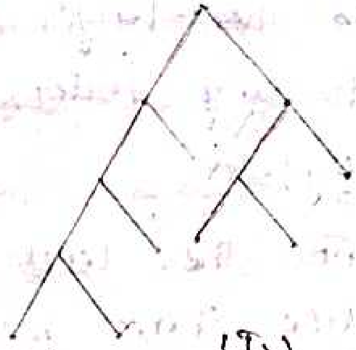
Balanced Tree :-

A rooted m-ary tree of Height h is balanced if all leaves are at levels h and h-1

Eg:

Which of the rooted tree shown in the figure are balanced.





The Height of rooted tree (T1) is 4. There are leaves in 3rd and 4th Levels.

∴ T1 is a balanced tree.

∴ T2 is Not balanced tree

Since the Height of T2 is 4 and there are some levels at Level 2 also.

Theorem :-

There are atmost m^h leaves in an m-ary tree of Height 'h'.

Proof :

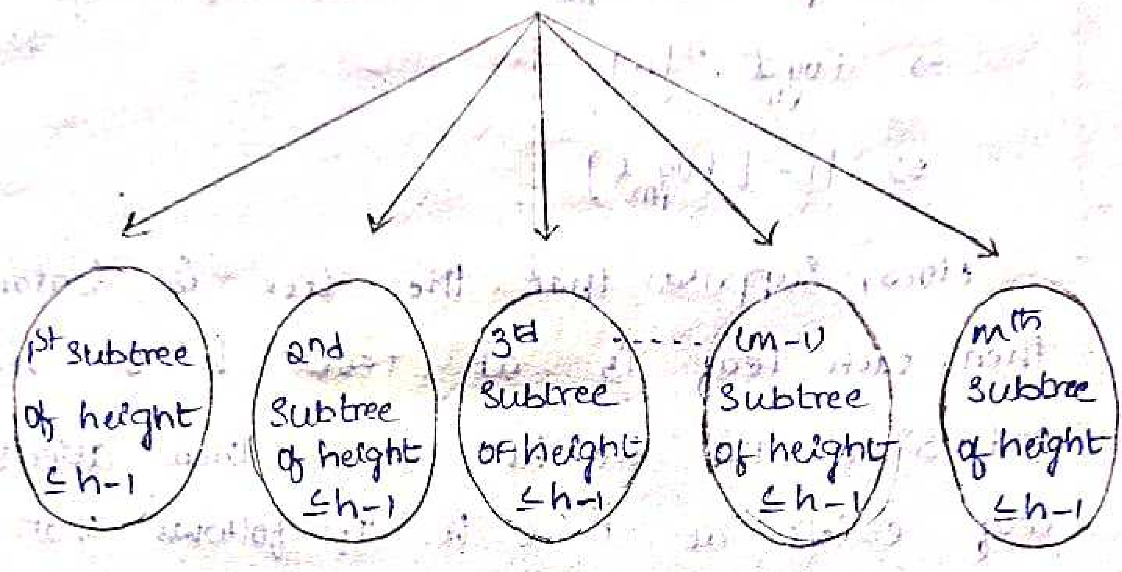
The proof uses Mathematical Induction on the height.

First consider a m-ary tree of height 1 This tree consists of a root with no more than m children. Each of which is a leaf Hence if there are no more than m leaves in an m-ary tree of height 1. This is the basic step of the Inductive argument.

We assume that the result is true for all m-ary tree of height less than h. This is the Inductive hypothesis

Let T be an m-array tree of

Height h. The leaves of T are the leaves of the subtrees of T obtained by deleting the edge from the root to each of the vertices at level 1 as shown in the figure.



Each of these subtrees has height less than or equal to $h-1$. So, by the inductive hypothesis each of these rooted trees has atmost m^{h-1} leaves. Since there are atmost m such subtrees, each with a maximum of m^{h-1} leaves.

Therefore, there are atmost $m \times m^{h-1} = m^h$ leaves in the rooted tree. This finishes the induction step as the theorem.

Theorem :-

If an m-ary tree of height 'h' has 'l' leaves then $h \geq \lceil \log_m l \rceil$. If the m-ary tree is full and balance then $h = \lceil \log_m l \rceil$

Proof :

We know that there are atmost m^h leaves in an m-ary tree of height h i.e., $l \leq m^h$.

Apply $\log_m$ on Both sides

$\Rightarrow \log_m l \leq \log_m m^h$

$\Rightarrow \log_m l \leq h \log_m m$

$\Rightarrow \log_m l \leq h \cdot 1$

$\Rightarrow h \geq \lceil \log_m l \rceil$

Now, Suppose that the tree is balanced then each leaf is at level h or $h-1$. and since, the height is h then atleast one leaf exists at level h. It follows that there must be more than m^{h-1} leaves and since $l \leq m^h$

$\Rightarrow m^{h-1} < l \leq m^h$

Taking $\log_m$ on both sides

$\Rightarrow \log_m m^{h-1} < \log_m l \leq \log_m m^h$

$\Rightarrow (h-1) \log_m m < \log_m l \leq h \log_m m$

$\Rightarrow (h-1) \cdot 1 < \log_m l \leq h \cdot 1$

$\Rightarrow h-1 < \log_m l \leq h$

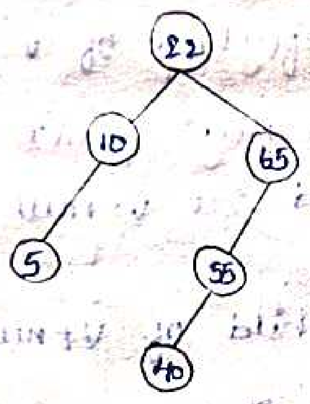
$\Rightarrow h = \log_m l$

Binary Search tree :-

The Binary Search tree is a binary tree in which each child of a vertex is designated as Left or Right child. The vertices are assigned keys so that the key of the vertex is both larger than the keys of all vertices in its left subtree and smaller than the keys of all vertices in its right subtree.

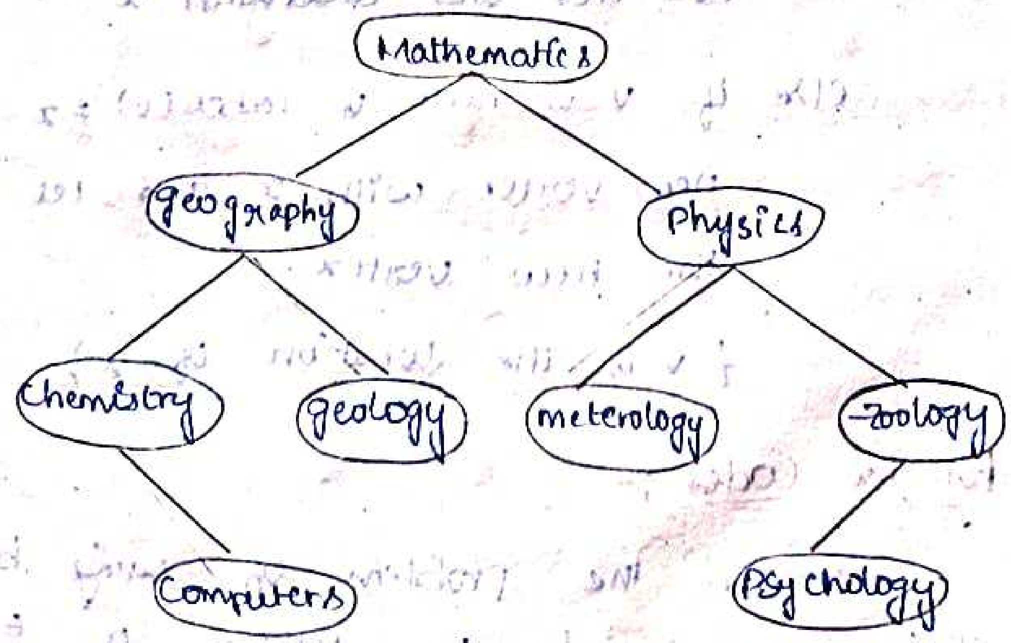
Eg: 1

Form a Binary Search tree for the following given numbers. 22, 10, 5, 65, 55, 40



Eg: 2 :

Form a Binary Search tree for the words mathematics, physics, Geography, zoology, meterology Geology, Psychology, chemistry and computers. Alphabetical order.



Algorithm of Binary Search Tree :-

Procedure Insertion (T : Binary Search tree ; x : item;
v : root (T))

{a vertex not in T has the value null}

while v ≠ null and label (v) ≠ x

begin

if x < label (v) then

if leftchild of v ≠ null then

v := left child of v

else add new vertex as a left child of v and set v = null

else

if rightchild of v ≠ null then v := right child of v.

else add new vertex as right child of v and set v := null.

end

if root of T := null then add a vertex v to the tree and label with x.

else if v is null or label (v) ≠ x then label new vertex with x and let v be the new vertex.

{v is the location of x}

Prefix codes :-

Consider the problem of using bit strings to encode the letters of English

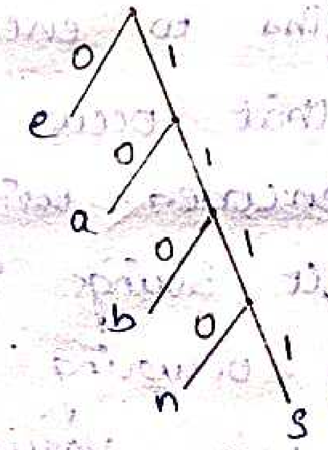
Alphabet. We can represent each letter with the bit string of length 5. The total number of bits used to encode data 5 times the number of characters in the text, when each character is encoded with 5 bits. We can reduce the required number of bits to encode a given text by using bit strings of different lengths to encode the letters.

Letters that occur more frequently should be encoded using sharp bit strings and larger bit strings should be used to encode rarely occuring letters. When letters are encoded using varying number of bits, some method must be used to determine where the bits for each character start and end.

One way to ensure that no bit string corresponds to more than one sequence of letters is to encode so, that the bit strings of a letter never occurs as the first part of the bit string for another letter. Codes with this property are called "Prefix Codes."

A prefix code can be represented by using a binary tree, where the characters are the labels of the leaves. The edges of the tree are labeled so that an edge leading to a left child is assigned '0' and an edge leading to Right child is assigned '1'. The bit strings used to encode a character is the sequence of labels of the edges in the unique path from the root to the leaf that has this character as its label

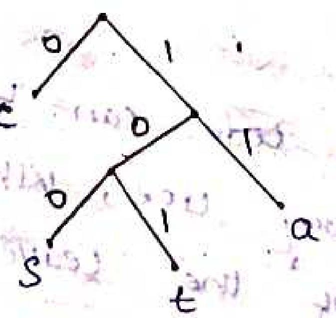
Eg:



e : 0
a : 10
b : 110
n : 1110
s : 1111

The bits for strings for each e, a, b, n, s are 0, 10, 110, 1110, 1111 respectively.

Construct the binary tree with the prefix codes a:11, c:0, ~~d~~ t:101, s:100



Huffman Coding :-

An algorithm that takes the frequencies of symbols in a string as input and produces as output the prefix code that encodes the string using fewest possible bits among all possible binary prefix codes for the symbols this Algorithm is known as "Huffman coding". It is a fundamental Algorithm for data compression

Algorithm :

procedure Huffman (C : symbol a_i with frequencies w_i, $i = 1, 2, \ldots n$)

F := Forest of n rooted trees, each consisting of vertex a_i and assigned weight w_i

while F is not a tree.

begin

Replace the rooted trees T and T' of least weights from F with $w(T) \geq w(T')$ and a tree having a new root that has T as its left subtree and T' as its right subtree. label the new edge to T with '0' and new edge to T' with '1'.

Assign $w(T) + w(T')$ as the weight of the new tree

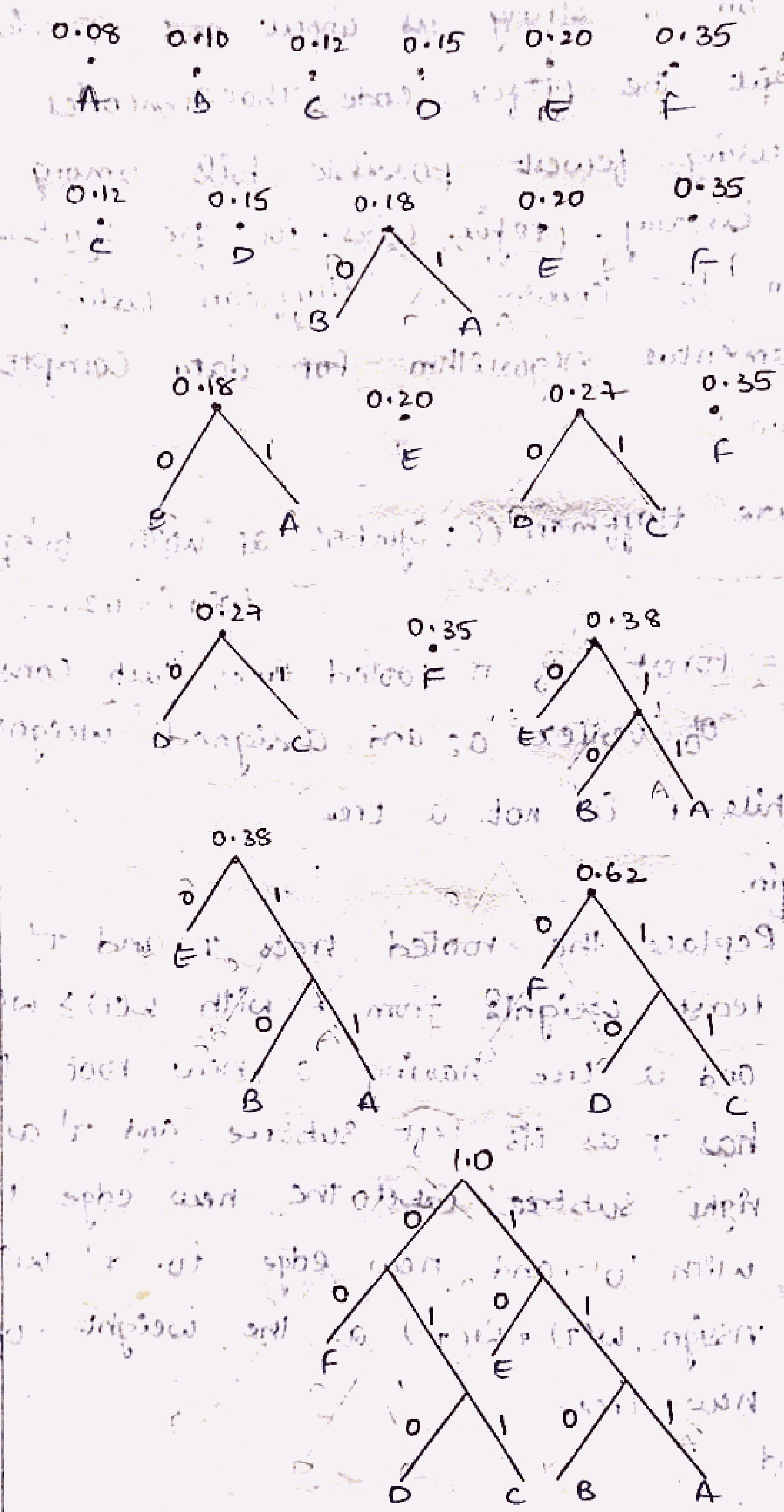
end

Eg :

use Huffman coding to encode the following symbols with the frequency listed below:

A = 0.08, B = 0.10, C = 0.12, D = 0.15, E : 0.20, F : 0.35

Find the average number of bits to use encode a character.



Therefore the bit string to represents the given characters are

A : 111 ; B : 110 ; C : 011 ; D : 010 ; E : 10 ; F : 00

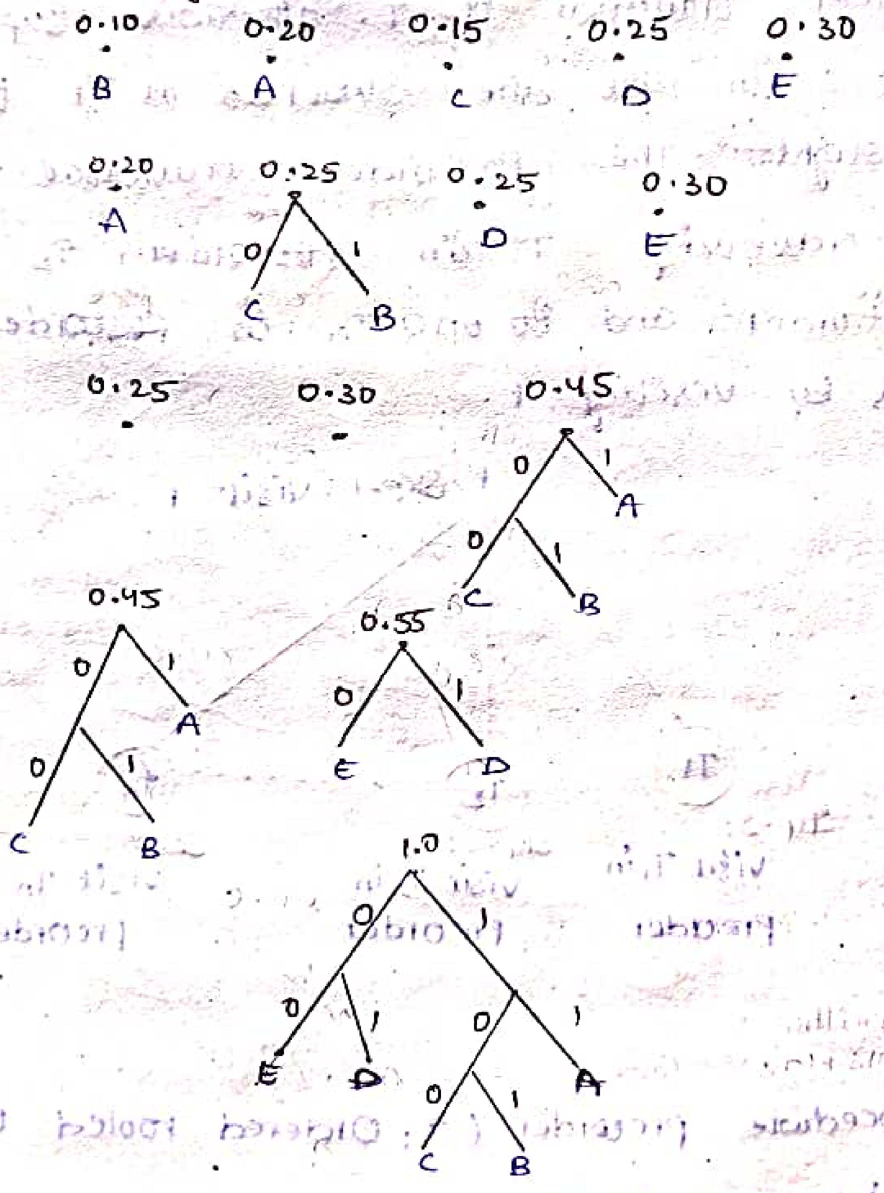
Average Number of bits required for representing a character $= 3(0.08) + 3(0.10) + 3(0.12) + 3(0.15) + 2(0.20) + 2(0.35)$

$= 2.45$

Use Huffman coding to encode the following Symbols

A: 0.20; B: 0.10; C: 0.15; D: 0.25; E: 0.30.

Find the average Number of bits required to encode the given symbols.



∴ The bit string & represent to the given character are A: 11, B: 101; C: 100; D: 01; E: 00

The average number of bits required

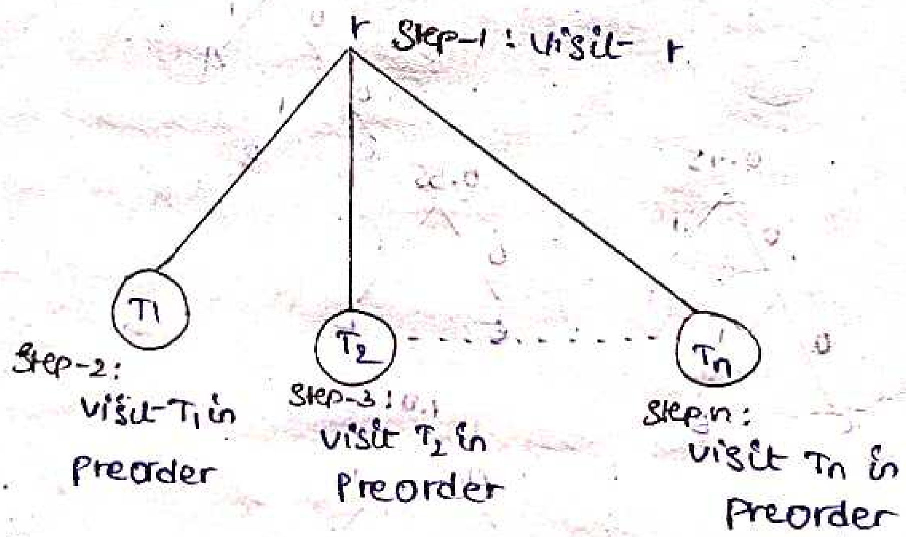
For representing a character $= 2(0.20) + 3(0.1) + 3(0.15) + 2(0.25) + 2(0.3)$
$= 2.25$

Tree Traversal Algorithm :-

Procedures for systematically visiting every vertex of an ordered rooted tree are called tree traversal algorithm.

1) Preorder Traversal :

Let 'T' be an ordered rooted tree with root r. If T consists only 'r' then r is the preorder traversal of T. Otherwise suppose that $T_1, T_2 \ldots T_n$ are the subtrees at r from left to right. The ~~Postorder~~ Pre traversal begins by traversing T_1 in ~~postorder~~ pre, T_2 in ~~postorder~~ pre and so on T_n in ~~postorder~~ pre and ends by visiting r.



Algorithm :

Procedure preorder (T: Ordered rooted tree)
 r := root of T
 list r
 For each child c of r from the left to Right

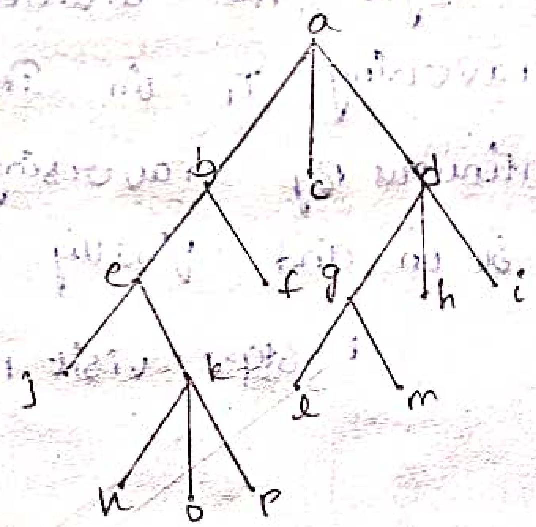
begin

T(c) := Subtree with 'c' as root Preorder (T(c))

end

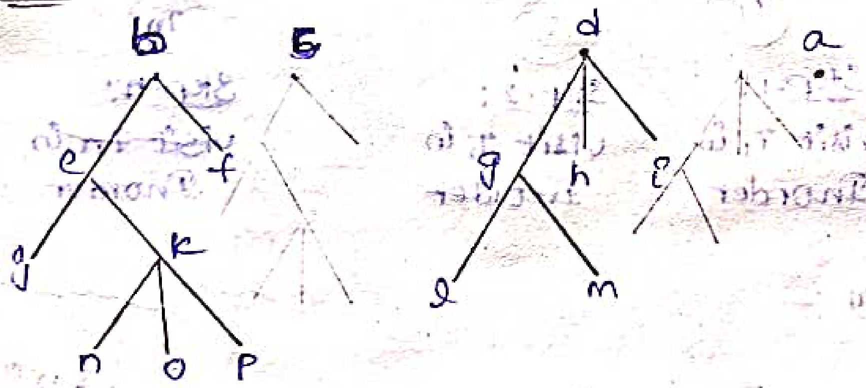
Eg :

1) In which order thus a preorder traversal visit the vertices in the ordered rooted root shown below :

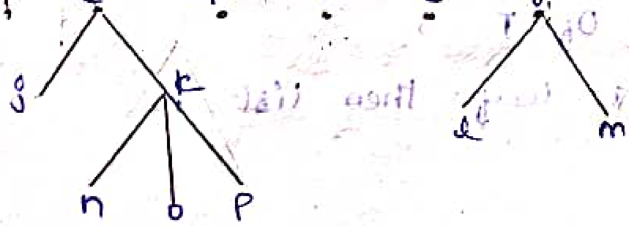


Post-order Solution

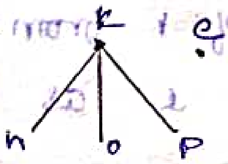
Step-1 :



Step-2 : e f b c g h i d a



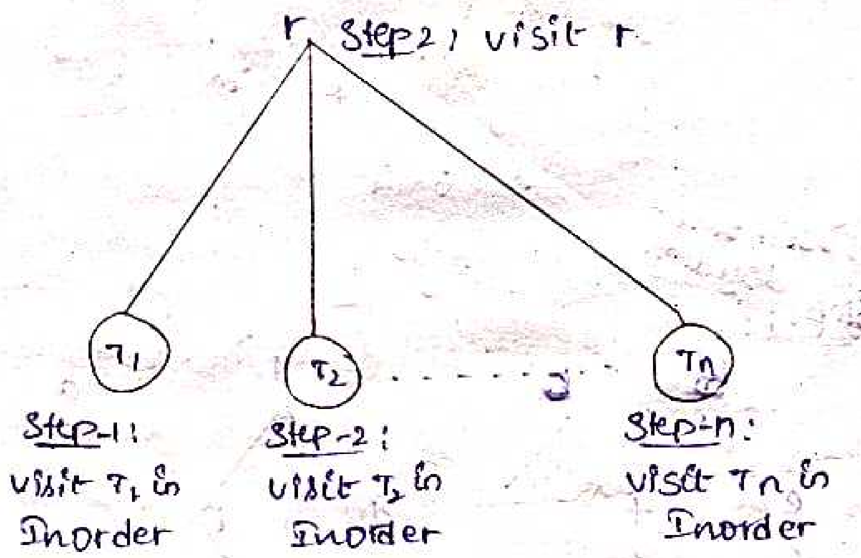
Step-3 : j k e f b c l m g h i d a



Step-4 : j n o p k e f b c l m g h i d a

Inorder Traversal :-

Let T be an ordered rooted tree with root r. If T consists only r then r is the inorder traversal of T. Otherwise suppose that $T_1, T_2 \dots T_n$ are the subtrees of r from left to right. The inorder traversal begins by traversing T_1 in inorder then visiting r. Continuous by traversing T_2 in inorder and so on and finally T_n in inorder.



Algorithm :-

```
Procedure Inorder(T: ordered rooted tree)
  r := root of T
  if r is a leaf then list r
  else
  begin
    l := first child of r from left to right
    T(l) := subtree with l as root
    Inorder(T(l))
    list r
```

for each child c of r except l from left to right

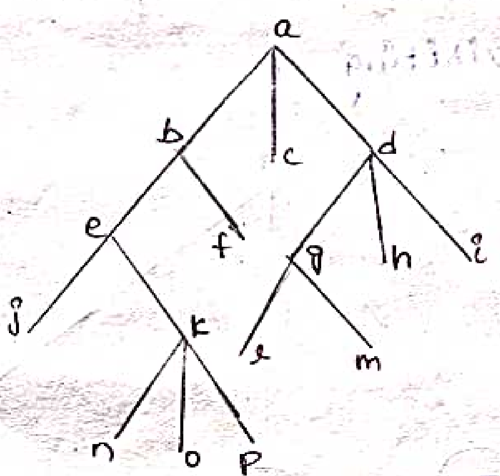
T(c) := Subtree with c as root

inorder (T(c))

end

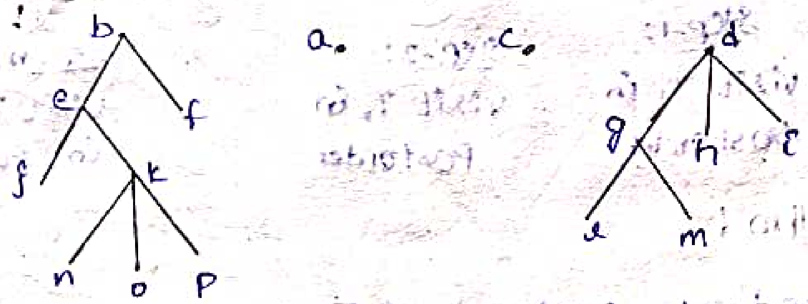
Example:

In which order the Inorder traversal visit the vertices in the ordered rooted tree.

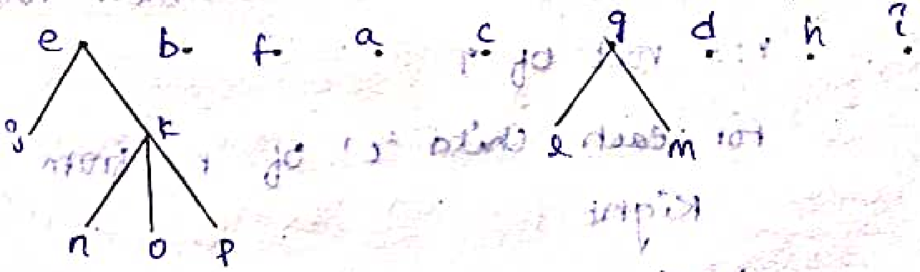


Sol

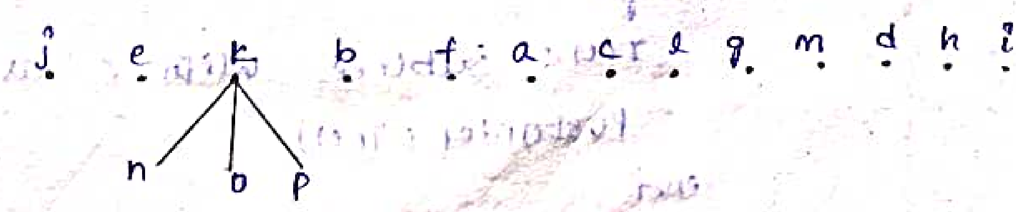
Step-1:



Step-2:



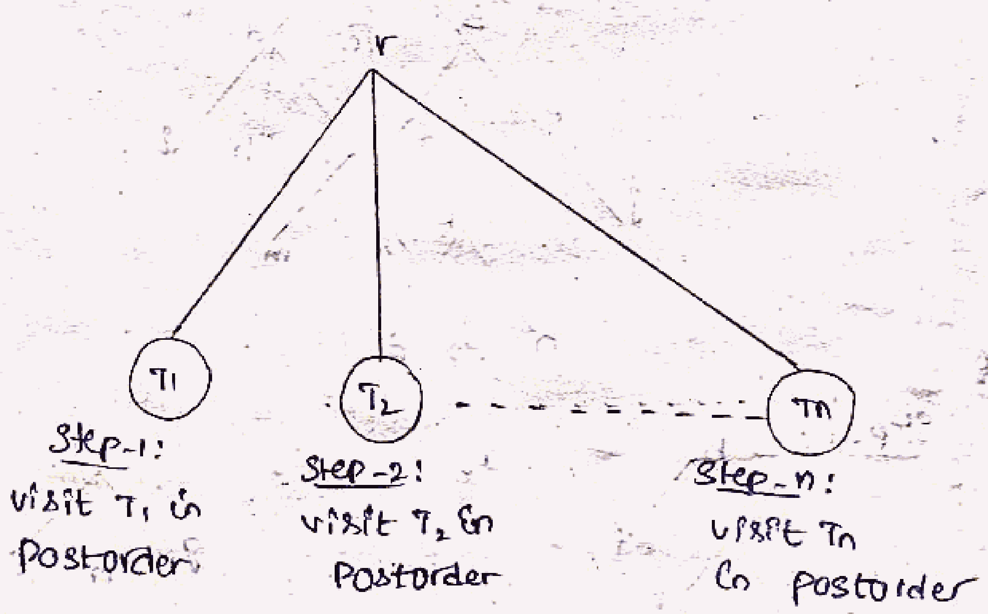
Step-3:



Step-4: j e n k o p b f a c l g m d h i

Postorder Traversal :-

Let T be an ordered rooted tree with root r. If T consists only r then r is the postorder traversal of T otherwise suppose that $T_1, T_2 \ldots T_n$ are the subtrees of r from left to right. The post order traversal begins by traversing T_1 in postorder T_2 in postorder ..... T_n in post order and ends by visiting r

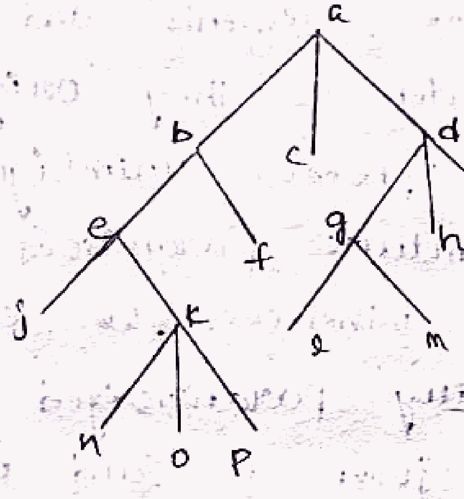


Algorithm :-

procedure postorder (T: ordered rooted tree)
 r := root of T
 for each child 'c' of r from left to Right
 begin
  T(c) := subtree with c as its root
  Postorder (T(c))
 end
 list r

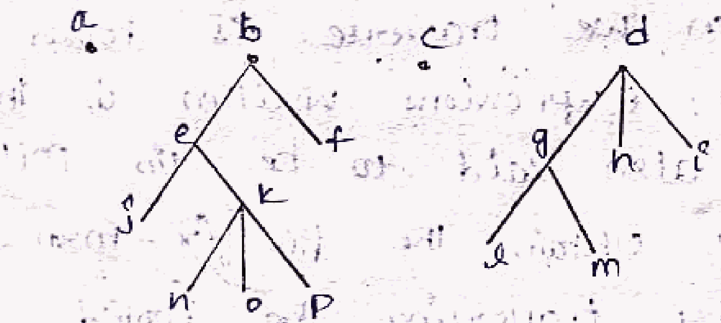
Eg:-

In which order does a postorder traversal visit the vertices of the ordered rooted tree T shown in the following figure.

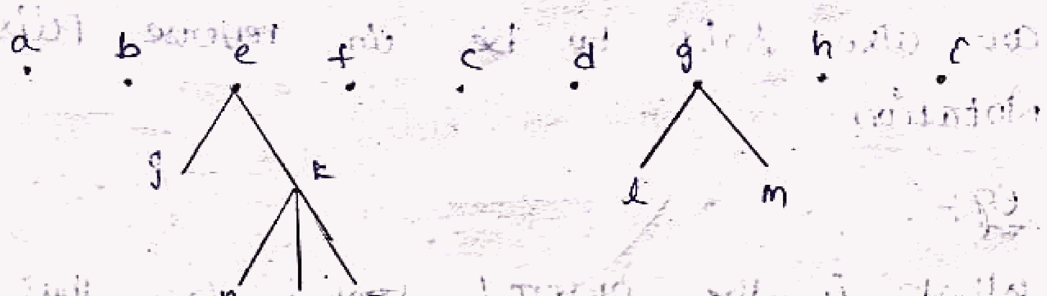


Sol: pre-order Solution

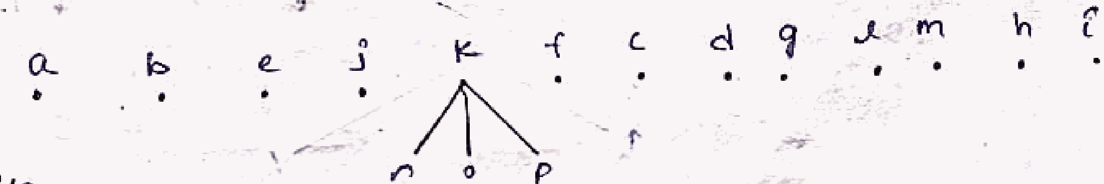
Step-1:



Step-2:



Step-3:



Step-4:

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

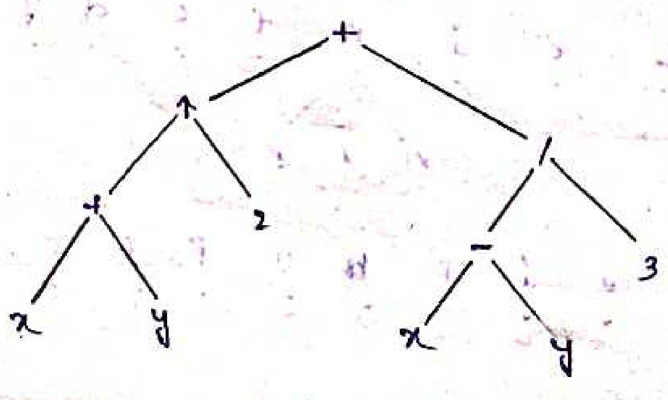
Infix, prefix and postfix Notations :

An Inorder traversal of the binary tree representing an Expression produces the original Expression with the elements and operations in the same order as they originally occur. To make such Expression unambiguous. It is Necessary to include paranthesis in the Inorder traversal whenever we encounter an operation. The Fully paranthesized Expression written in this form is said to be in infix form.

We obtain the prefix form of an Expression when we traverse its rooted tree in preorder. Expressions written in this form are also said to be in polish Notation. We obtain the postfix form of an Expression by traversing the rooted tree in postorder. Expressions written in this form are also said to be in reverse polish Notation.

Eg :-

1) What is the ordered rooted tree that represents the Expression $(((x+y)\uparrow 2)+(x-y)/3)$



The prefix Notation of the given Expression is

$+ \uparrow + x\, y\, 2\, 1 - x\, 4\, 3$

The Postfix Notation of the given Expression is

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

2) What is the value of the prefix Expression

(a) $+ - * 2\, 3\, 5 / \uparrow 2\, 3\, 4$

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

$= + - * 2\, 3\, 5\, \underline{/ 8\, 4}$

$= + - \underline{* 2\, 3}\, 5\, 2$

$= + \underline{- 6\, 5}\, 2$

$= \underline{+ 1\, 2}$

$= 3$

(b) $- * 2 / 8\, 4\, 3$

$- * 2\, \underline{/ 8\, 4}\, 3$

$= - \underline{* 2\, 2}\, 3$

$= \underline{- 4\, 3}$

$= 1$

3) What is the value of the postfix Notation

(a) $7\, 2\, 3 * - 4 \uparrow 9\, 3 / +$

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

$= 7\, \underline{2\, 3 *} - 4 \uparrow 3 +$

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

$= \underline{1\, 4 \uparrow}\, 3 +$

$= \underline{1\, 3 +}$

$= 4$

(b) 5 2 1 - - 3 1 4 + + *

$5\ 2\ 1 - - 3\ \underline{1\ 4\ +} + *$

$= 5\ 2\ 1 - - \underline{3\ 5\ +} *$

$= 5\ \underline{2\ 1\ -} - 8\ *$

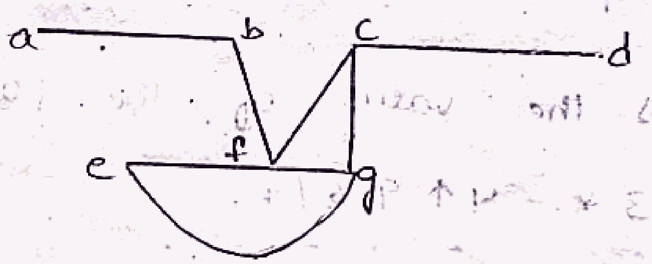
$= \underline{5\ 1\ -}\ 8\ *$

$= \underline{4\ 8\ *}$

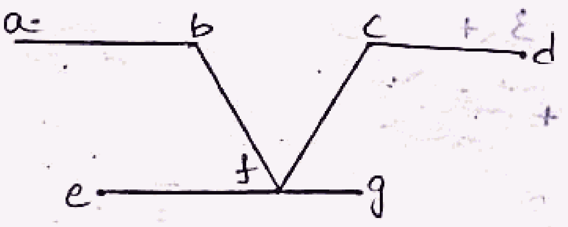
$= 32$

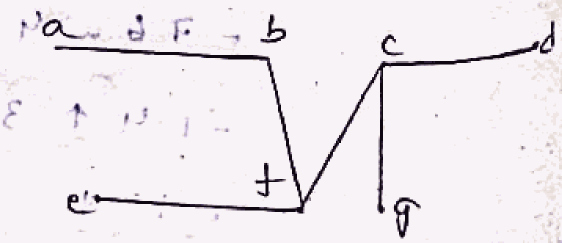
Spanning Trees :-

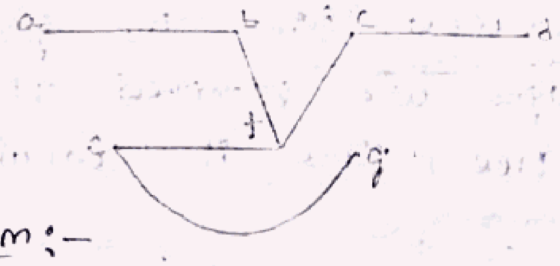
Let G be a simple graph, a spanning tree of G is a subgraph of G i.e a tree containing every vertex of G. A simple graph with a spanning tree must be connected, since there is a path in the spanning tree between any two vertices. The converse is also true i.e, Every connected simple graph has a spanning tree.



The possible spanning trees for the given graph are







Theorem :-

A Simple Graph is connected iff and only if it has a Spanning tree

Proof :-

⇐ let G be a graph and it has a Spanning tree T. Since T contains every vertex of G Further more there is a path in T between every vertex any two of its vertices.

Since T is a Subgraph of G, there is a path in G between any two of its vertices.

Hence G is Connected.

⇒ Conversely Suppose that G is connected. let G is not a tree, it must contain a Simple circuit. Remove an edge from one of these simple circuits. The resulting Sub-graph has one fewer edge but still contains all vertices of G and is connected. If this sub graph is not a tree, it has a simple circuit so as behave remove an edge i.e., in a simple circuit. Repeat this process until no simple circuits Remain. This is possible because there are only a finite number of edges in the graph.

A tree is produced, since the graph space connected as edges are removed. This tree is a spanning tree, since it contains every vertex of G.

Therefore, a connected simple graph as a spanning tree.

Breadth First Search (BFS) :-

We can produce a spanning tree of the simple graph by the use of Breadth first search. We orbitarly choose a root from the vertices of the graph. Then add all edges incident to this vertex. The new vertices added at this space become the vertices at level 1 in the spanning tree.

In orbitary order add each edge incident to the vertex at level 1 as long as it doesnot produce a simple circuit. Follow the same ~~prod~~ procedure until all vertices of the graph has been added.

Algorithm :-

Procedure BFS (G: connected graph with vertices $v_1, v_2, \ldots, v_n$)

T := Tree consisting only of vertex v_1

L := Empty list

Put v_1 in the list L of unprocessed vertices

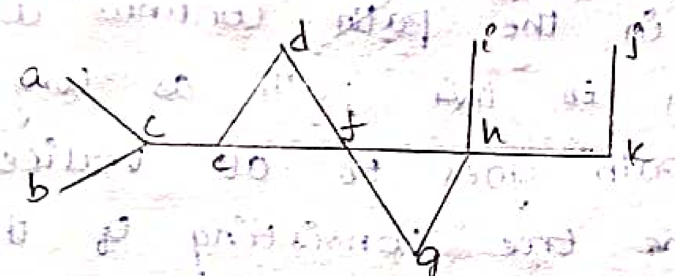
while L is not empty

begin

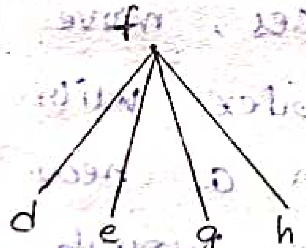
remove the first vertex v from L

for each neighbour w of v
 if w is not in L and not in T then
 begin
 add w to the end of the list L
 add w and edge {v,w} to T
 end
end

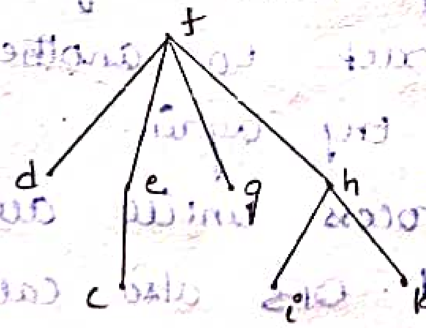
Use BFS to find a spanning tree for graph given below.



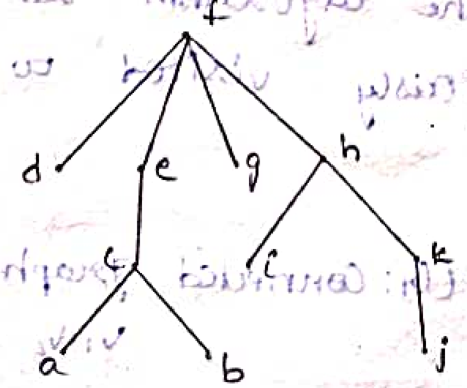
Sol:-



— Step-1



— Step-2



— Step-3

Depth First Search (DFS) :-

We can build a spanning tree for a connected simple graph using DFS. We arbitrarily choose a vertex of the graph as the root. Form a path starting at this vertex by successively adding vertices and edges, where each new edge is incident with the last vertex in the path and a vertex not already in the path. Continue adding vertices and edges to this path as long as possible. If the path goes to all vertices of the graph, the tree consisting of this path is a spanning tree. If the path doesnot go through all vertices, move back to the next to last vertex with this path and if possible form a new path starting at this vertex, passing through vertices that were not already visited. If this cannot be done move back to another vertex in the path and try again.

Repeat this process untill all the vertices are added. DFS also called "Back Tracking", since the algorithm returns to vertices previously visited to add paths

Algorithm :-

Procedure DFS (G: Connected graph with vertices $v_1, v_2, \dots, v_n$)

T := Tree consisting only of vertex v_1

visit (v_1)

procedure visit (v: vertex of G)

for each vertex w adjacent to v and not yet in T

begin

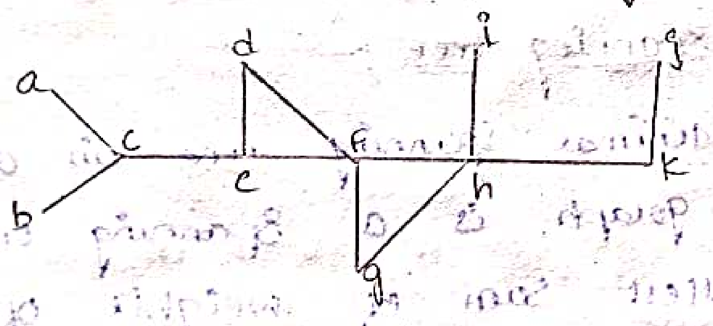
add vertex w and edge {v, w} to T

visit (w)

end

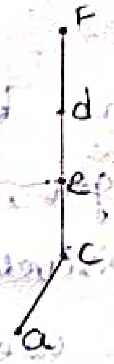
Eg:

use DFS to find the spanning tree for the graph shown in the following figure.

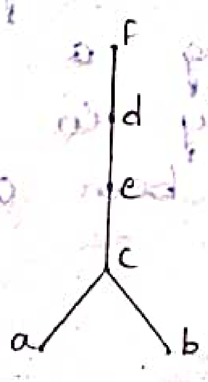


Sol: Step-1 :

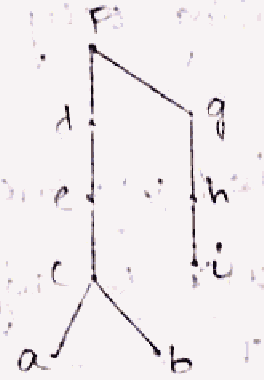
Step-2 :



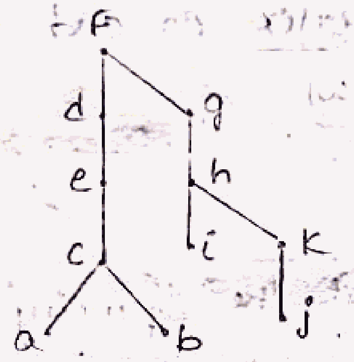
Step-3 :



Step-4:



Step-5:



Minimal Spanning tree :-

A Minimal spanning tree in a connected weighted graph is a spanning tree that has smallest sum of weights of its edges.

Prim's Algorithm :-

This Algorithm begins by choosing any edge with smallest weight, putting it into the spanning tree. Successively add to the tree, the edges of minimum weight that are incident to a vertex already in the tree and not forming a simple circuit with does edges already in the tree. Stop when n-1 edges have been added.

Algorithm :

procedure prims (G: weighted connected undirected graph with 'n' vertices)

T:= a minimum weight edge

for i:= 1 to n-2 do

begin

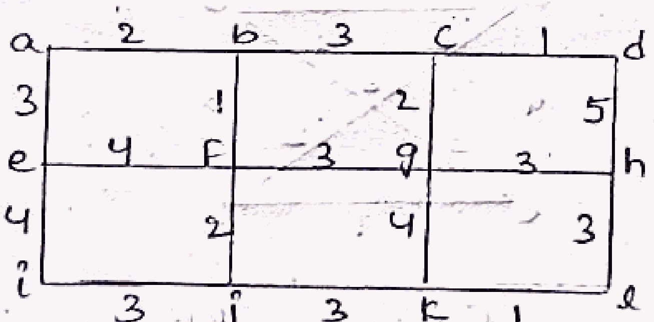
e:= edge of minimum weight incident to a vertex in T and not forming a circuit in T iff added to T

T: = T∪{e}

end { T is a minimal spanning tree of G}

Eg:

use prim's algorithm to find a minimal spanning tree in the graph given below.

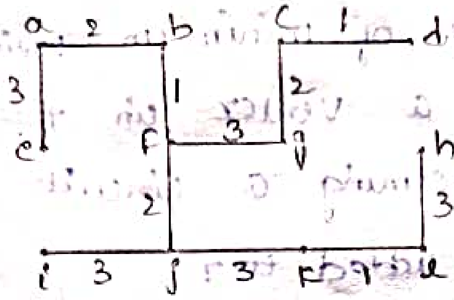


38

choice	edge	weight
1.	{c,d}	1
2	{c,g}	2
3	{g,f}	3
4	{f,b}	1
5	{f,g}	2
6	{b,a}	2

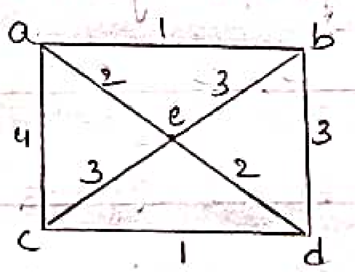
7	{a, e}	3
8	{j, i}	3
9	{j, k}	3
10	{k, l}	1
11	{l, h}	3

<u>Graph</u>



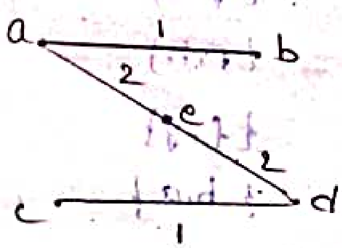
The above spanning tree is the minimal spanning tree of the given graph is the minimal weight 24

2. Use Prim's Algorithm to find a minimal spanning tree in the graph given below.



<u>choice</u>	<u>edge</u>	<u>weight</u>
1	{a, b}	1
2	{a, e}	2
3	{e, d}	2
4	{d, c}	1

<u>Graph</u>:



The above Spanning tree is the Minimal Spanning tree with minimal weight 6

Kruskal's Algorithm :-

The Algorithm begins by choosing an edge in the graph with minimum weight. Successively add edges with minimum weight that donot Form a Simple circuit with does edges already choosen. Stop after n-1 edges has been Selected.

Algorithm :

Procedure Kruskal (G : weighted connected un-directed graph with n vertices)

T := empty graph

for i := 1 to n-1 do

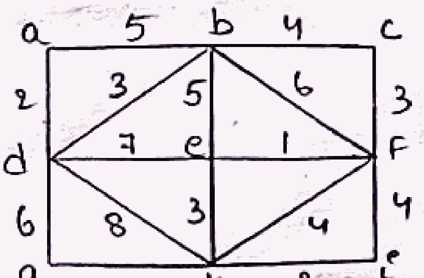
begin

e := any edge in G with smallest weight that doesnot Form a Simple circuit when added to T

T := T ∪ {e}

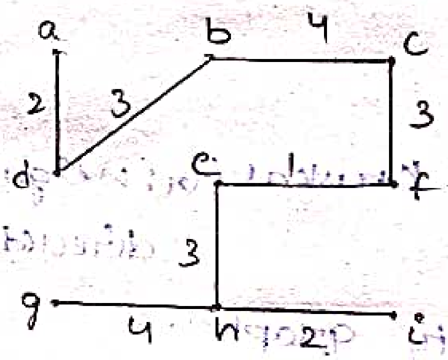
end {T is the minimal Spanning tree of G}

1. Use Kruskal's algorithm to find a minimal Spanning tree in the weighted graph shown below.



Choice	edge	weight
1	{e,f}	1
2	{a,d}	2
3	{h,i}	2
4	{b,d}	3
5	{c,f}	3
6	{e,h}	3
7	{b,c}	4
8	{g,h}	4

Graph :-



The above spanning tree is the minimal spanning tree with minimal weight.

Modeling Computation

Unit - 4

Languages and Grammars :-

A vocabulary or alphabet Σ is a finite non-empty set of elements called symbols. A word or sentence over Σ is a string of finite length of elements of Σ. The empty string is denoted by λ, is the string containing no symbols. The set of all words over Σ is denoted by Σ^*.

A language over Σ is a subset of Σ^*

Phrase structured Grammar :

A phrase structured Grammar $G = (V, T, P, S)$ where V is a vocabulary, a subset T of V consisting of terminal elements, a set of productions P and a start symbol S from V. The $V-T$ is denoted by 'N', non terminal set. Every production in P must contain atleast one non-terminal on its left side.

Eg: let $G = (V, T, P, S)$

where $V = \{a, b, A, B, S\}$

$T = \{a, b\}$

S is the starting symbol, and

$P = \{S \to ABa, A \to BB, B \to ab, AB \to b\}$

Let $G = (V, T, P, S)$ be a phrase structured grammar let $w_0 = lz_0r$ and $w_1 = lz_1r$ be string over V. If $z_0 \to z_1$ is a production of G, we say that w_1 is directly derivable from w_0 and we write $w_0 \underset{G}{\Rightarrow} w_1$. If $w_0, w_1 \ldots w_n$ are strings over V such that $w_0 \underset{G}{\Rightarrow} w_1 \underset{G}{\Rightarrow} \cdots \underset{G}{\Rightarrow} w_n$ then we say that w_n is derivable from w_0 and we write w_0 derives $w_0 \overset{*}{\underset{G}{\Rightarrow}} w_n$. The sequence of steps used to obtain w_n from w_0 is called a derivation.

The language generated by G denoted by L(G), is the set of all strings of terminals that are derivable from starting symbol S. In other words $L(G) = \{w \in T^* / S \overset{*}{\Rightarrow} w\}$

Eg:

1) Let G be a Grammar with vocabulary. $V = \{S, A, a, b\}$, The set of terminals $T = \{a, b\}$, starting symbol S and productions $P = \{S \to aA, S \to b, A \to aa\}$ what is L(G).

Since $S \to b \in G$ then $S \Rightarrow b$

then $b \in L(G)$

Since $S \to aA \in G$ then the derivation is

$S \Rightarrow aA$

$\Rightarrow aaa \quad (\because A \to aa)$

The $S \overset{*}{\Rightarrow} aaa$ then $aaa \in L(G)$

$\therefore L(G) = \{b, aaa\}$

2) Let G be a Grammar with vocabulary $V = \{S, 0, 1\}$, set of terminals $T = \{0, 1\}$, an

Starting symbol S and and production P

$P = \{S \to 11S, S \to 0\}$.

Sol: Since $S \to 0 \in G$ then $S \Rightarrow 0$

$\therefore 0 \in L(G)$

Suppose we start with $S \to 11S$ the derivation proceeds in the following manner.

$$S \Rightarrow 11S$$
$$\Rightarrow 1111S$$
$$\vdots$$
$$\overset{n}{\Rightarrow} 1^{2n}S$$
$$\Rightarrow 1^{2n}0 \quad (\because S \to 0 \in G)$$

$\Rightarrow S \overset{*}{\Rightarrow} 1^{2n}0$

That is we apply $S \to 11S$ n numbers of times then we derive $1^{2n}S$, Inorder to find the terminal string. finally we use $S \to 0$ once.

$\therefore L(G) = \{1^{2n}0 / n \geq 0\}$

3) Give a phrase structured Grammar that generates the set $\{0^n 1^n / n \geq 0\}$

Sol: The given language $L = \{0^n 1^n / n \geq 0\}$

$= \{\lambda, 01, 0^2 1^2, 0^3 1^3 \ldots\}$

The required phrase structured grammer $G = (V, T, P, S)$ where

$V = \{S, 0, 1\}$

$T = \{0, 1\}$

S is the starting symbol

$P = \{S \to 0S1, S \to \lambda\}$

By using the above specified grammar suppose we start with $S \to \lambda$ then $\lambda \in L(G)$. Otherwise we start with $S \to 0S1$ and applying repeatdly n times, finally apply $S \to \lambda$ to get the terminal string $0^n 1^n$

i.e., $S \Rightarrow 0S1$

$$\Rightarrow 0^2 S 1^2$$

$$\vdots$$

$$\Rightarrow 0^n S 1^n$$

$$\Rightarrow 0^n 1^n$$

$$\therefore S \overset{*}{\Rightarrow} 0^n 1^n$$

4) Find a phrase structured grammar to generate the set $\{0^m 1^n / m, n \geq 0\}$

Sol:- The Required phrase structured grammar $G = (V, T, P, S)$ where

$V = \{S, A, 0, 1\}$

$T = \{0, 1\}$

S is the starting symbol

$P = \{S \to 0S, S \to 1A, A \to 1A, A \to \lambda, S \to \lambda\}$

Types of Phrase Structured Grammar:-

Phrase structured Grammar can be classified according to the types of productions that are allowed.

→ A Type-0 Grammar has no more restriction on its production. So every production is a type-0 production.

→ A Type-1 Grammar can have productions of the form $w_1 \to w_2$ where the length of w_2 is greater than or equal to w_1 ($w_2 \geq w_1$)

→ A Type-2 Grammar can have productions of the form $w_1 \to w_2$ where the w_1 is a single symbol that is not a terminal symbol

→ A Type-3 Grammar can have productions of the form $w_1 \to w_2$ with $w_1 = A$ and $w_2 = aB$ or $w_2 = a$ where a is a terminal and A, B are non-terminals.

A Type-0 Grammar is also called as universal Grammar. A Type-1 Grammar is also called as context Sensitive Grammar. A Type-2 Grammar is called as a context free Grammer. finally type-3 Grammar is also called as a regular Grammar.

Derivation Tree :-

A Derivation tree in the language generated by a context free Grammar can be represented Graphically using an ordered Rooted tree, called derivation Tree or parse tree. The root of this tree Represent the starting symbol, the internal vertices of the tree Represent the non-terminal symbols that occur in the derivation. The leaves of the tree Represent the terminal symbol If the productio that occur in the derivation.

$A \to w$ arrives in the derivation, where w is the word, the vertex that Represents A has children vertices that represents each symbol in w. In the order from left to Right.

Eg:

Test whealther cbab belongs to the language generated by the Grammar $G = (V, T, P, S)$ when $V = \{a, b, c; A, B, C, S\}$, $T = \{a, b, c\}$, S is the starting symbol and $P = \{S \to AB, A \to Ca, B \to Ba, B \to Cb, B \to b, C \to Cb, C \to b\}$.

Sol:- Since G is the given Grammar and Cbab is the string to be derived the derivation is as follows.

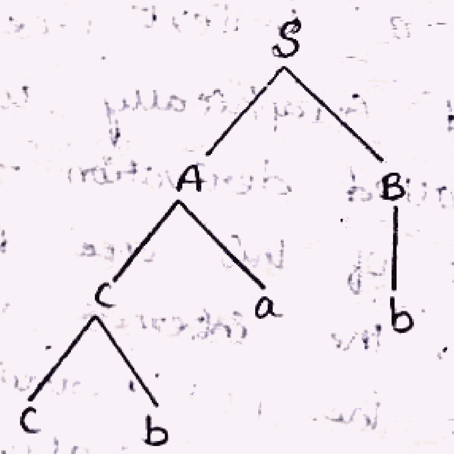
$S \Rightarrow AB \quad (\because S \to AB)$

$\Rightarrow CaB \quad (\because A \to Ca)$

$\Rightarrow CbaB \quad (\because C \to Cb)$

$\Rightarrow Cbab \quad (\because B \to b)$

Its derivation tree is as shown below.



TOP - down Parsing :-

The way of approach the problem is to

begin with S and attempt to derive the required terminal string using a series of production is called top-down parsing

Bottom-up parsing :

Bottom-up parsing is an approach in which we work backward i.e., we begin with the terminal string and constructing derivation by using a sequence of production to reach the starting symbol.

Eg: $S \Rightarrow AB \Rightarrow Ab \Rightarrow Cab \Rightarrow Cbab$ (Top-down)

$S \Leftarrow AB \Leftarrow Ab \Leftarrow Cab \Leftarrow Cbab$ (Bottom-up)

Backus - Naur Form (BNF) :-

It is used to specify the syntactic Rules of many computer Languages. The productions in the type-2 Grammar have a single non-terminal symbol as their left hand side Instead of using all the productions seperately we can combine all those with the same non-terminal on the left hand side into one statement. Instead of using symbol '$\rightarrow$' in a production we use the symbol '::='. We enclose all non-terminal symbol in < > and we list all the Right hand sides of the production in the same statement, seperating them '|' (bars).

Eg: $A \rightarrow Aa$, $A \rightarrow a$, $A \rightarrow AB$ can be combined into a single statement is

<A> ::= <A> a | a | <A> <B>

1. Give Backus-Naur Form for the production of signed integers in decimal notation.

Sol: <Signed Integer> ::= <sign><integer>

<sign> ::= + | -

<integer> ::= <digit> | <digit><integer>

<digit> ::= 0|1|2|3|4|5|6|7|8|9

2. By using the above Grammar construct a derivation tree for +135

Sol: <signed-integer> ⇒ <sign> <integer>

⇒ + <integer>

⇒ + <digit> <integer>

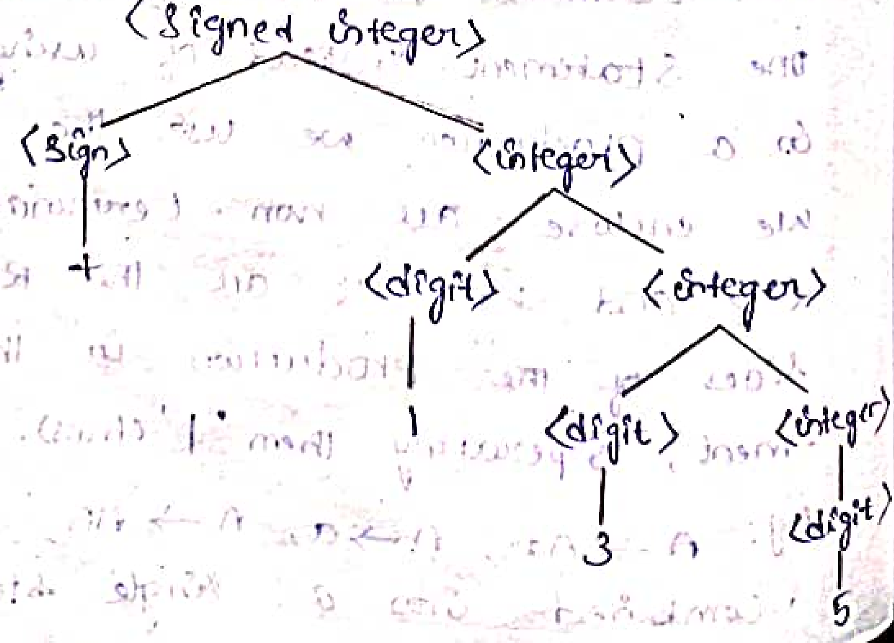
⇒ + 1 <integer>

⇒ +1 <digit> <integer>

⇒ +13 <integer>

⇒ +13 <digit>

⇒ +135



Finite State Machines with output :-

All the versions of a Finite State Machine include a finite set of States, with a designated starting state, an input alphabet and a transaction function that assigns a next State to every state and input pair. Finite state Machines are used extensively in computer science applications and data Networking.

A Finite State Machine $M = (S, I, O, f, g, S_0)$ consists of a finite set S of states, a finite input alphabet I, a transaction function f, a finite output Alphabet O, the output function g that assigns to each state and input pair an output, and an initial state S_0.

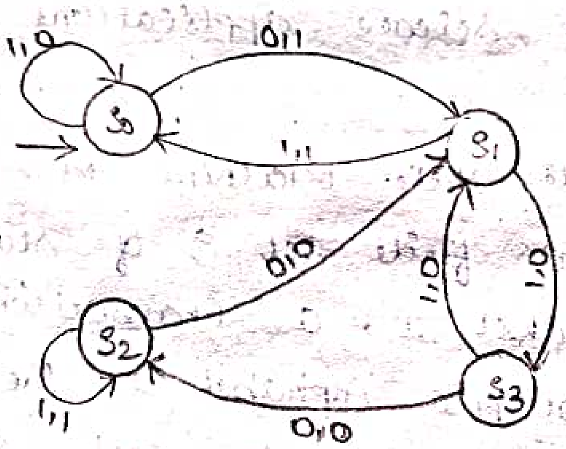
We can use a state table to represent the values of f and g for all pairs of states, output and input. Another way to represent a finite state Machine is a state diagram which is a directed graph with labelled edges. In this diagram each state is represented by a circle, Arrows labeled with the input and output pairs.

Eg :

1. Construct a (or) Transition state diagram for the finite state Machine with the state table shown below.

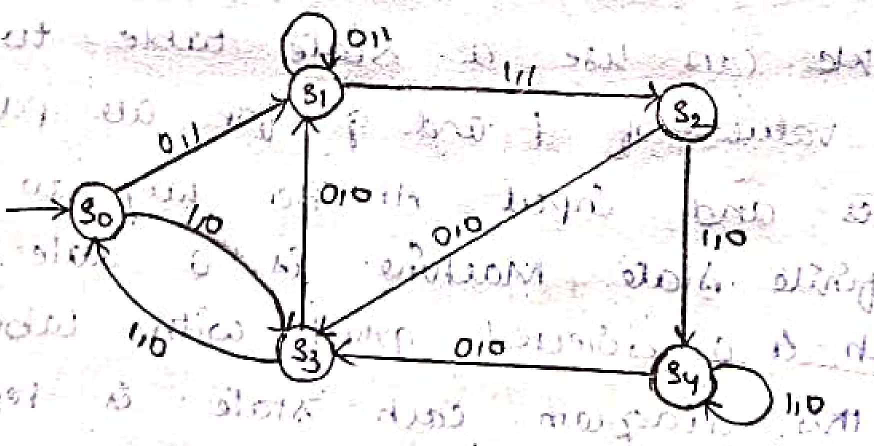
	f		g	
	Input		Input	
	0	1	0	1
S_0	S_1	S_0	1	0
S_1	S_3	S_0	1	1
S_2	S_1	S_2	0	1
S_3	S_2	S_1	0	0

Sol:-



2. Construct the state table for the finite state machine shown in the state diagram

(i)



Sol:-

State	f		g	
	Input		Input	
	0	1	0	1
S_0	S_1	S_3		0
S_1	S_1	S_2	1	1
S_2	S_3	S_4	0	0
S_3	S_1	S_0	0	0
S_4	S_3	S_4	0	0

(ii) Find the output string generated by the Finite State Machine given above if the input string is 101011

$$S_0 \xrightarrow[0]{1} S_3 \xrightarrow[0]{0} S_1 \xrightarrow[1]{1} S_2 \xrightarrow[0]{0} S_3 \xrightarrow[0]{1} S_0 \xrightarrow[0]{1} S_3$$

The output string is 001000

Types of Finite State Machines :-

In 'Mealy Machine' the output correspond to transmission between the states i.e., depending on the state and input where as in 'More machine' the output is determined the state only.

Concateration :-

Suppose that A and B are subsets of V where V is the vocabulary, the concatinatio of A and B denoted by AB, is the set of all strings of the form xy where x is a string in A and y is a string in

$$AB = \{xy / x \in A, y \in B\}$$

Eg: $A = \{0, 11\}$, $B = \{1, 10, 110\}$

$AB = \{01, 010, 0110, 111, 1110, 11110\}$

Find A^3

$A^3 = A^2 \cdot A$

$A^2 = \{0, 11\} \cdot \{0, 11\}$

$= \{00, 011, 110, 1111\}$

$A^3 = \{00, 011, 110, 1111\} \cdot \{0, 11\}$

$A^3 = \{000, 0110, 1100, 11110, 0011, 01111, 11011, 111111\}$

Kleene closure :

Suppose that A is a subset of V^* ($A \subseteq V^*$) then the Kleene closure of A denoted by A^*, is a set of strings of concantination of orbitary strings from A i.e., $A^* = \bigcup_{k=0}^{\infty} A^k$

Supposet let $A = \{0\}$ then $A^* = \{\lambda, 0, 00, 000, 0000 ....\}$

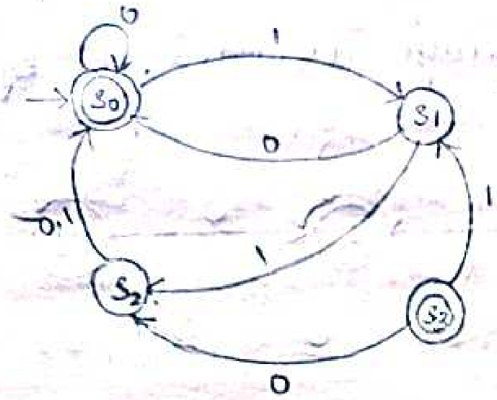
Finite State Automata : (DFA)

A Finite State Automata $M = (S, I, f, s_0, F)$ consists of a finite set S of states, a finite input Alphabet I, a transistion function f that assigns a next state to a state, input pair. so, $f: S \times I \to S$, an initial state s_0 and $F \subseteq S$ consisting of final states. we can represent the finite state automata using either state tables or state diagrams. The final states are indicated in state diagrams with double circle.

Eg:

Construct the state diagram for the finite Automata $M = (S, I, f, s_0, F)$ where $S = \{s_0, s_1, s_2, s_3\}$, $I = \{0, 1\}$, $F = \{s_0, s_3\}$ and the transistion function f is given in the following table

State	f Input 0	f Input 1
s_0	s_0	s_1
s_1	s_0	s_2
s_2	s_0	s_0
s_3	s_2	s_1

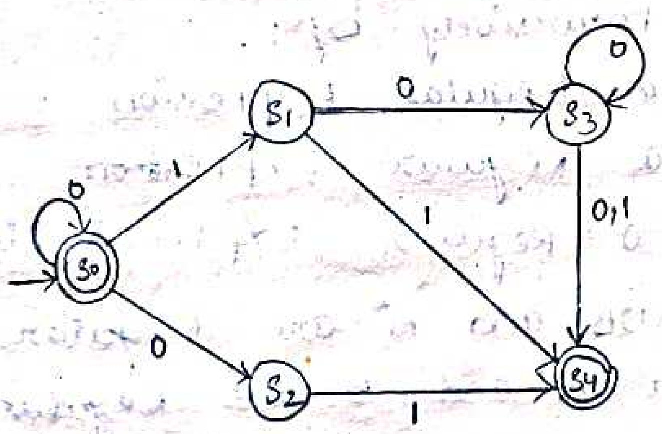


Non - Deterministic Finite State Automata (NDFA)

A NDFA $M = (S, I, f, S_0, F)$ consists of a set S of states, an Input Alphabet I, a transition Function f that assigns a set of states to each pair of state and Input, A starting state S_0 and $F \subseteq S$ consisting of Final states.

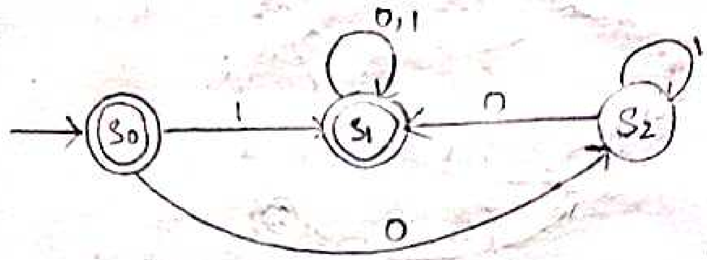
Eg:

Find the state table for the NFA given below.



	f	
State	Input 0	Input 1
S_0	S_0, S_2	S_1
S_1	S_3	S_4
S_2	—	S_4
S_3	S_3, S_4	S_4
S_4	—	—

2. Find the language recognised by the given DFA.



Sol: The language recognised by this DFA consists of the string λ, the string starting with 1 followed by as many number of 0's and 1's. The string starting with 0 followed by as many number of ones followed by 0 followed by as many number of 0's and 1's.

Regular Expression :

The Regular Expression over a set I is defined recursively by :

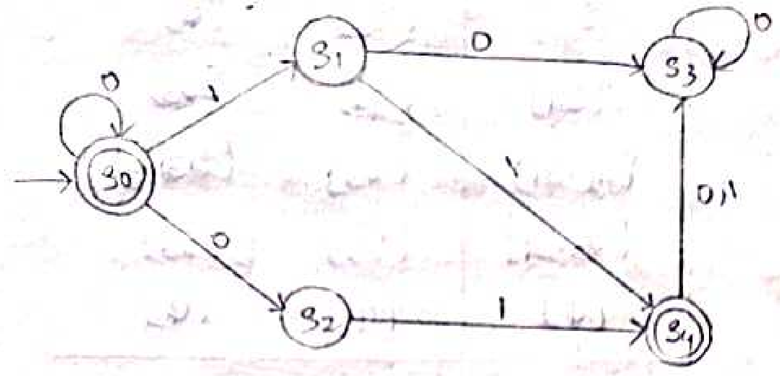
1) ϕ is a Regular Expression
2) λ is a Regular Expression
3) x is a Regular Expression, if $x \in I$
4) (AB), $(A \cup B)$ and A^* are Regular Expressions whenever A and B are Regular Expressions

Eg: 10^* : A 1 followed by any number of 0's.

$(10)^*$: Any Number of copies of 10

$0 \cup 01$: The string 0 (or) the string 01

1) Find DFA that recognise the same language as the NDFA given below.



Sol: The given NFA, $M = (S, I, f, S_0, F)$ where $S = \{S_0, S_1, S_2, S_3, S_4\}$, $I = \{0,1\}$, S_0 is the initial state, $F = \{S_0, S_4\}$, f is shown in the below state table.

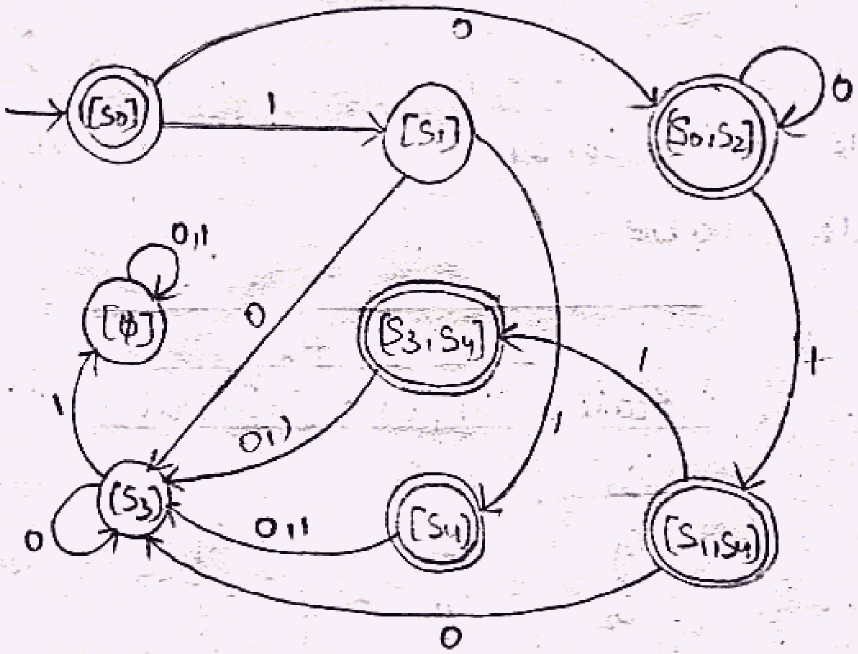
State	f: Input 0	Input 1
S_0	S_0, S_2	S_1
S_1	S_3	S_4
S_2	ϕ	S_4
S_3	S_3	ϕ
S_4	S_3	S_3

The Equivalent DFA that recognises the same language as with the above NFA is $M' = (S', I, f', S_0', F')$ where $S' = \{[S_0], [S_1], [S_0, S_2], [S_3], [S_4], [S_1, S_4], [S_3, S_4], [\phi]\}$, $S_0' = \{[S_0]\}$, $F' = \{[S_0], [S_0, S_2], [S_4], [S_1, S_4], [S_3, S_4]\}$, f' is shown in the following state table.

State	f': input 0	input 1
$[S_0]$	$[S_0, S_2]$	$[S_1]$
$[S_1]$	$[S_3]$	$[S_4]$
$[S_0, S_2]$	$[S_0, S_2]$	$[S_1, S_4]$

[S_3]	[S_3]	[ϕ]
[S_4]	[S_3]	[S_3]
[S_1, S_4]	[S_3]	[S_3, S_4]
[S_3, S_4]	[S_3]	[S_3]
[ϕ]	[ϕ]	[ϕ]

Its state diagram is drawn below.



Language Recognition :-

Regular Expressions :

The regular Expressions over a set 'I' are designed recursively by the symbol ϕ is a regular expression.

* The symbol λ is a regular Expression.

* The Symbol x is a Regular Expression whenever $x \in I$.

* The symbols (AB), (AUB) and A^* are regular Expressions. Whenever A and B are regular expressions.

Eg !

What are the strings in the regular sets specified by the regular Expression 10^*, $(10)^*$, 0U01, 0U(0U1)* and (0* 1)*

The regular sets represented by the expressions are given below

Expression	Strings
10^*	a '1' followed by any number of 0's
$(10)^*$	Any number of copies of '10' (including the null string)
0U01	The string '0' or the string '01'
$0U(0U01)^*$	Any string starting with '0' (or) a '0' followed by any number of 0's or 1's (or) 0's and 1's
$(0^* 1)^*$	Any string ending with '1'.

Turning Machines :-

A turning Machines $T = (S, I, f, S_0)$ consists of a finite set S of states an alphabet I containing the blank symbol B, a partial function f from $S \times I$ to $S \times I \times \{R, L\}$ and a starting state S_0

Let I be a subset of an alphabet V. A turning Machine $T = \{S, T, f, S_0\}$ recognizes a string x in $V^* \Leftrightarrow$ T starting in the initial position when x is written on the tape.

Turning Machine is an abstract Machine i.e, they are not real machines. They are developed before the computers are dollar. In 1938, they are developed by "Alen quering" because his name after that three scientists.

The turning Machine is 7-tuple

→ $M = (Q, \Sigma, \delta, q_0, F)$

$M = (Q, \Sigma, \delta, q_0, B, \Pi, F)$

Where Σ : Set of input symbols

δ : Transaction function

$\delta : Q \times \Pi \rightarrow Q \times \Pi \times \{L, R\}$

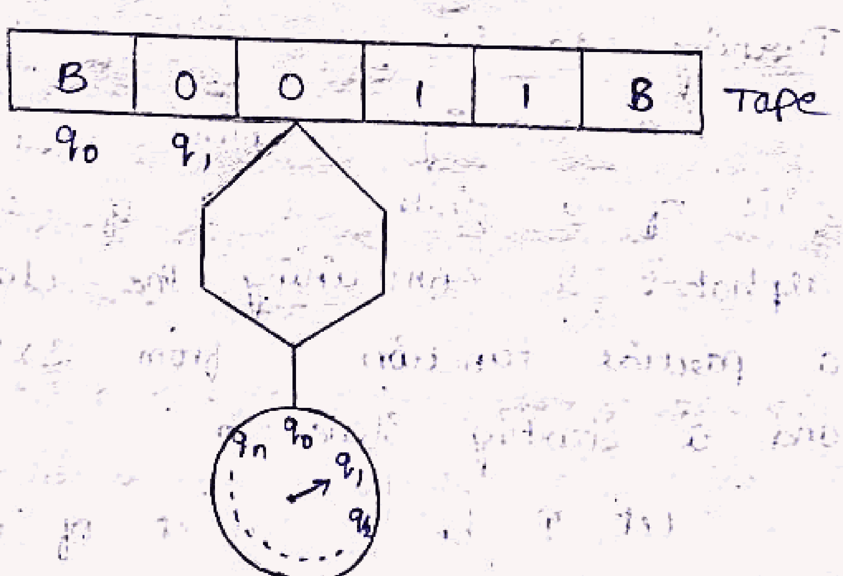
q_0 : Initial state

F : Final state

B : Blank space

Π : Set of tape symbols.

<u>Initial description of turning Machines</u> :



The initial description consists of the state with one tape symbol on the left hand side. If we apply transition function then on the right hand side we will go to new state (or) same state by taking one tape symbol and its also takes the direction either left or Right.

Construction of turning Machine :-

Eg: construct a turning Machine for a function. 0 function is given by $z(x) = 0$

Assumptions :-

1) Starting with q_0, if q_0 reads blank space, state will be changed to q_1 and blank space remains same and moves to right.

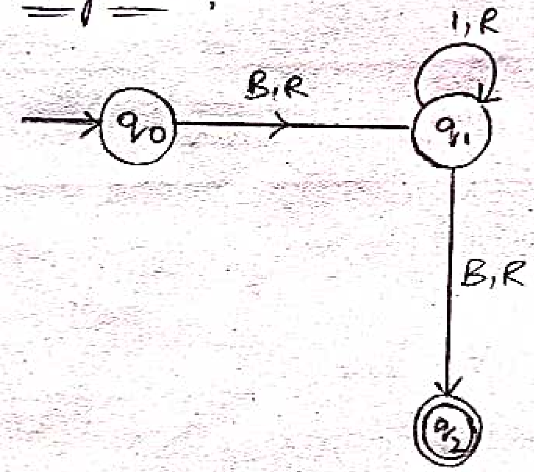
2) When q_1 reads '1', then the state remains

same and '1' is replaced by blank space, moves to right.

A transition table is

States	1	B
q_0	-	(q_1, B, R)
q_1	(q_1, B, R)	(q_2, B, R)
q_2	-	-

Transition diagram:

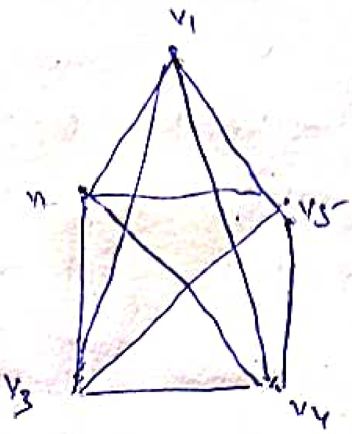


Concatenation of two strings:

Suppose w_1 and w_2 are two strings then the concatenation of w_1 and w_2 are given by $Conc(w_1, w_2) = w_1 w_2$.

Eg: If $w_1 = 11$, $w_2 = 111$ then $Conc(w_1, w_2)$?

Sol:- $Conc(w_1, w_2) = w_1 w_2 = 11111$



K_5