Toppersnotes

Unleash the topper in you

UGC-NET

COMPUTER SCIENCE AND APPLICATIONS

National Testing Agency (NTA)

PAPER 2 || VOLUME – 1

SIERRA INNOVATIONS

UGC NET PAPER – 2

COMPUTER SCIENCE AND APPLICATIONS

I UNIT

Discrete Structures and Optimization

Mathematical Logic

Logic which uses a symbolic language to express its principles in precise and unambiguous terms is known as mathematical logic.

One reason for this is that all efforts at the verification of algorithms inevitably the notation and methods of logic. Logic, among other things, have provided the theoretical basis for many areas of computer science such as digital logic design, automata theory and computability, and artificial intelligence etc.

<u>Propositions:-</u>

A number of words making a complete grammatical structure having a sense and meaning and also meant an assertion in logic or mathematics is called a sentence.

This assertion may be of two types- declarative and non-declarative.

A proposition is a declarative sentence that is either true or false but not both. A proposition may have truth value T or F, T stands for the case when the proposition is true and F stands for the case when the proposition is false.

Propositional logic is the simplest form of logic where all the statements are made by propositions.

The purpose of using propositional logic is to analyze a statement, individually or compositely.

Propositional logic is also called boolean logic as it works on 0 and 1.

For example, "two plus one equals three" and "Three plus three equals seven" are both statements,

the first because it is true and the second because it is false. Similarly "$x+y > 1$" is not a statement because for some values of x and y the sentence is true, whereas for others it is false.

For instance, if $x = 1$ and $y = 2$, the sentence is true, if $x = -3$ and $y = 1$, this is false.

The truth or falsity of a statement is called its truth value. Since only two possible truth values are admitted this logic is sometimes called two-valued logic.

Questions, exclamations and commands are not propositions.

It is customary to represent simple statements by letters p,q,r,··· known as proposition variables. Propositional variables can only assume two values, true or false.

In Propositional logic , we use symbolic variables to represent the logic, and we can use any symbol for a representing a proposition, such as A, B, C, D, P, Q, R etc.

Propositional logic consists of an object, relations or function and logical connectives.

<u>Compound Proposition:-</u>

A proposition consisting of only a single propositional variable or a single propositional constant is called an atomic (primary, primitive) proposition or simply proposition; that is, they can not be further subdivided.

In other word, a proposition is called a simple proposition if it cannot be reduced into another simpler proposition.

A proposition obtained from the combinations of two or more propositions by means of logical operators or connectives of two or more propositions or by negating a single proposition is referred to molecular or composite or compound proposition.

Connectives

The words or phrases (or symbols) used to form compound propositions are called connectives. These connectives are also called logical operators.

There are five basic connectives called

- Negation
- Conjunction
- Disjunction
- Implication or Conditional, and
- Equivalence or Biconditional

Symbol used	Connective word	Nature of the compound statement formed by the connective	Symbolic form	Negation
~,¬,N	not	Negation	~p	~(~p)=p
∧	and	Conjunction	p∧q	(~p)∨(~q)
∨	or	Disjunction	p∨q	(~p)∧(~q)
⇒,→	if, ⋯ then	Implication(or) Conditional	p⇒q	p∧(~q)
⇔,<−>	if and only if	Equivalence(or) Bi-conditional	p⇔q	p∧(~q)]∨[q∧(~p)]

Proposition and Truth Tables

For Conjunction

Conjunction of propositions p and q is denoted by p ∧ q.

p ∧ q is read as "p and q".

The proposition p ∧ q is true whenever both p and q are true, otherwise p ∧ q is false.

p	q	p ∧ q
T	T	T
T	F	F
F	T	F
F	F	F

For Disjunction

Disjunction of propositions p and q is denoted by p ∨ q.

p ∨ q is read as " p or q".

The statement p ∨ q is true whenever either p or q is true or both p and q are true; otherwise, p ∨ q is false.

p	q	p ∨ q
T	T	T

T	F	T
F	T	T
F	F	F

For Negation

Negation of a proposition p is denoted by ~p is read as "not p ".

p	**~p**
T	F
F	T

For Implication / Conditional Proposition

If p and q are proposition, the compound proposition "if p then q" denoted by $p \Rightarrow q$ is called a conditional proposition or implication and the connective is the conditional connective. The proposition p is called antecedent or hypothesis, and the proposition q is called the consequent or conclusion.

p	**q**	**$p \rightarrow q$**
T	T	T
T	F	F
F	T	T
F	F	F

For Biconditional Statement

If p and q are statements, then the compound statement p if and only if q, denoted by $p \Leftrightarrow q$ is called a biconditional statement and the connective if and only if is the biconditional connective. The biconditional statement $p \Leftrightarrow q$ can also be stated as "p is a necessary and sufficient condition for q" or as "p implies q and q implies p".

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

Negation of Compound Statements:-

Negation of Conjunction: A conjunction $p \wedge q$ consists of two sub-statements p and q both of which exist simultaneously. Therefore, the negation of the conjunction would mean the negation of at least one of the two sub-statements.

Thus, we have "the negation of a conjunction $p \wedge q$ is the disjunction of the negation of p and the negation of q". Equivalently, we write

$\sim(p \wedge q) \equiv \sim p \vee \sim q$.

In order to prove the above equivalence, we prepare the following table.

Negation of Disjunction: A disjunction $p \wedge q$ consists of two sub-statements p and q which are such that either p or q or both exist.

Therefore, the negation of the disjunction would mean the negation of both p and q simultaneously.

p	q	p∧q	~(p∧q)	~p	~q	~p∨~q
T	T	T	F	F	F	F
T	F	F	T	F	T	T
F	T	F	T	T	F	T
F	F	F	T	T	T	T

The negation of a disjunction p∧q is the conjunction of the negation of p and the negation of q. Equivalently, we write

~(p∨q)≡~p∧~q

In order to prove the above equivalence, we prepare the following table.

p	q	~p	~q	~p∧~q	p∧q	~(p∧q)
T	T	F	F	F	T	F
T	F	F	T	F	T	F
F	T	T	F	F	T	F
F	F	T	T	T	F	T

Negation of Implication: If p and q are two statements, then

~(p⇒q)≡p∧~q

In order to prove the above equivalence, we prepare the following table.

p	q	p⇒q	~(p⇒q)	~q	p∧~q
T	T	T	F	F	F
T	F	F	T	T	T
F	T	T	F	F	F
F	F	T	F	T	F

Negation of Biconditional: If p and q are two statements, then

~(p⇔q) ≡ p⇔~q ≡ ~p⇔q

Law of algebra of propostions :-

The laws are listed in the following table

Law	**Expression(s)**
Idempotent Laws	$p \vee p = p, p \wedge p = p$
Associative Laws	(i) $p \vee (q \vee r) = (p \vee q) \vee r$ (ii) $p \wedge (q \wedge r) = (p \wedge q) \wedge r$
Distributive Laws	(i) $p \vee (q \wedge r) = (p \vee q) \wedge (p \vee r)$ (ii) $p \wedge (q \vee r) = (p \wedge q) \vee (p \wedge r)$
Commutative Laws	(i) $p \vee q = q \vee p$ (ii) $p \wedge q = q \wedge p$
Involution Law	$\sim (\sim p) = p$
Identity Laws	(i) $p \vee \mathrm{F} = p, p \wedge \mathrm{T} = p$ (ii) $p \vee \mathrm{T} = \mathrm{T}, p \wedge \mathrm{F} = \mathrm{F}$
Complement Laws	(i) $p \vee \sim p = \mathrm{T}, p \wedge \sim p = \mathrm{F}$ (ii) $\sim \mathrm{T} = \mathrm{F}, \sim \mathrm{F} = \mathrm{T}$
De Morgan's Laws	(i) $\sim (p \vee q) = \sim p \wedge \sim q$ (ii) $\sim (p \wedge q) = \sim p \vee \sim q$

For example,

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

and

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

hold good and are called absorption laws.

These can be derived in the following manner.

Let us consider:

$p \vee (p \wedge q) \quad = (p \wedge T) \vee (p \wedge q) \qquad [\because p \wedge T = p]$

$\qquad = p \wedge (T \vee q) \qquad$ [Distributive law]

$\qquad = p \wedge T \qquad [\because T \vee q = T]$

$\qquad = p$

De'Morgan Laws:-

(i) $\sim (p \wedge q) = \sim p \vee \sim q$

(ii) $\sim (p \vee q) = \sim p \wedge \sim q$

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

From the table we see that

$-(p \wedge q) \equiv \sim p \vee \sim q$

Similarly, you can verify that

$\sim (p \vee q) \equiv \sim p \wedge \sim q$

Sets and Relations

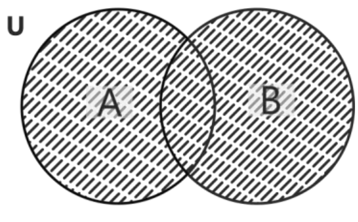
Set operations :-

Set operations include set union, set intersection, set difference, complement of set and cartesian product.

Union of sets :-

The union of sets A and B (denoted by A ∪ B) is the set of elements that are in A, in B, or in both A and B.

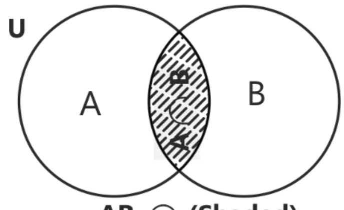
A∪B = {x : x ∈ A or x ∈B}



A∪B (Shaded)

Intersection of sets:-

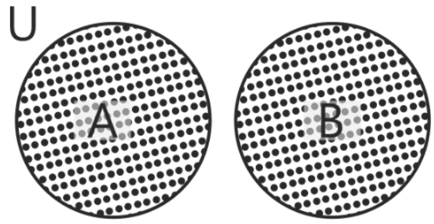
The intersection of sets A and B (denoted by A ∩ B) is the set of elements which are in both A and B.



AB ∩ (Shaded)

if A and B have no common element, then $A \cap B = \emptyset$.

In this case, the two sets A and B are called disjoint sets.

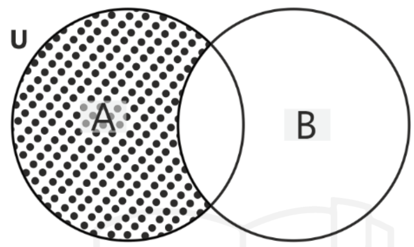


$\mathbf{A \cap B = \phi}$

it is obvious that $A \cap B \subseteq A, A \cap B \subseteq B$.

<u>Difference of sets:-</u>

The set difference of sets A and B (denoted by A -B) is the set of elements that one only in A but not in B.



<u>Complement of a set</u> :- If U be the universal set of a set A, then the set of all those elements in U which are not members of A is called the Compliment of A, denoted by A^C or A'

A′ (Shaded)

clearly,

A' = U - A

Cartesian product / Cross Product :-

if we take two sets A = {a,b} and B= {1, 2}

⇒ the cartesian product of A and B is written as

A *B = {(a,1),(a,2),(b,1),(b,2)}

⇒ the cartesion product of B and A is written as

B * A = {(1,a),(1,b), (2,a),(2,b)}

<u>Laws of the Algebra of sets:-</u>

these laws can be directly used to prove different propositions on set theory.

1. **Idempotent laws:** $\boldsymbol{A \cup A = A, A \cap A = A}$
2. **Commutative laws:** $\boldsymbol{A \cup B = B \cup A, A \cap B = B \cap A}$
3. **Associative laws:** $\boldsymbol{A \cup (B \cup C) = (A \cup B) \cup C, A \cap (B \cap C) = (A \cap B) \cap C}$
4. **Distributive laws :** $\boldsymbol{A \cup (B \cap C) = (A \vDash B) \cap (A \cap C), A \cap (B \cup C) = (A \cap B) \cup (A \cap C)}$
5. **Identity laws:** $\boldsymbol{A \cup \phi = A, A \cup U = U}$
6. $\boldsymbol{A \cap U = A, A \cap \phi = \phi}$

7. **Complement laws:** $A \cup A' = U, A \cap A' = \phi$
8. $(A')' = A, U' = \phi, \phi' = U$
9. **De Morgan's laws :** $(A \cup B)' = A' \cap B'$
10. $(A \cap B)' = A' \cup B'$.

Counting, Mathematical Induction and Discrete Probability

Fundamental Principle of Counting:-

The fundamental counting principle can be used to determine the
number of possible outcomes for compound events.

It has two basic rules:

1. **Sum rule:-** If an event can occur in m different ways and another event can occur in n different ways and if these two events can not be done at the same time i.e they are independent, then either of the two events can occur in (m + n) ways. The Sum rule of counting can also be generalised to any number of finite events.

   Let $A_1, A_2, \dots\dots\dots, A_n$ be disjoint sets, then the number of ways to choose any element from one of these sets is

   $|A_1 \cup A_2 \cup \dots\dots\dots \cup A_n| = |A_1| + |A_2| + \dots\dots\dots + |A_n|$

2. **Product rule:-** If an event can occur in m different ways and if following it, another event can occur in n different ways, then in the given order, both the events can occur in the given order, both the events can occur in the m × n, i.e. mn ways.

   This principle can be generalised to any finite number of events.

   Let $A_1, A_2, \dots\dots\dots, A_n$ be finite sets, then the number of ways to choose one element from set in the order $A_1, A_2, \dots\dots\dots A_n$ is,

   $|A_1 \times A_2 \times \dots\dots\dots \times A_n| = |A_1| \times |A_2| \times \dots\dots\dots \times |A_n|$

   where, $|A_i|$ (i = 1, 2, … … … n) denotes the number of elements in the set A_i.

The pigeonhole principle:-

If there are n+1 objects and n boxes, then there is at least one box with two or more objects.

Assume that we have n+1 objects and every boxes has at most one object.

Then the total number of objects is n, which is a contradiction because there are n+1 objects. Hence if n+1 objects are placed into n boxes, then there is at least one box containing two or more objects.

The generalised pigeonhole principle states that ''If N objects are placed into K boxes, then there is at least one box containing at least [N / K] objects.''

Example:-

there are 10 Pigeons and 9 pigeon holes when pigeons fly to home then one of them have to stay two in one hole.

Permutations :-

The different arrangements that can be made with a given number of things taking some or all of them of a time are called permutation.

A permutation is an ordered combination.

For example, the permutation of the letters a, b, c taken two at a time are ab, ba, ac, ca, bc, cb.
The symbol ^nP_r or P (n, r) is used to denote the number of
permutations of n distinct things taken r at a time.

Permutation of n-distinct objects:

Theorem: The number of different permutations of n- distinct objects taken r at a time is given by

^np_r = n (n–1) (n–2) … … … [n–(r–2)]

$^np_r = \frac{n!}{(n-r)!}$ [r <=n]

Note:

1) n! = n(n - 1)!

= n(n - 1)(n - 2)!

= n(n - 1)(n - 2)(n - 3)! etc.

2) In this case, repetition is not allowed.

So, we can fill:

- the first place in n ways,
- the second place in (n - 1) ways,
- the third place in (n - 2) ways,
- and so on,
- until the r^{th} place, which can be filled in [n - (r - 1)] ways.

Thus, the total number of ways to fill r places is:

n × (n - 1) × (n - 2) × … × [n - (r - 1)] = ^nP_r

3) If repetition is allowed, then each of the r-places can be filled in n ways.

So, we can fill:

Hence, all the r places can be filled in:

n × n × n × … × n = n^r ways (r times)

4) The number of permutations of n things taken all at a time is n!

By definition, ^nP_n = n(n – 1)(n – 2)…2.1 = n!

5) We have, $^np_r = \frac{n!}{(n-r)!}$

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

$= \frac{n!}{0!}$

or, n! $= \frac{n!}{0!}$

∴ 0! $= \frac{n!}{n!} = 1$

Thus, 0! = 1

Example 1: Prove the Following Relations

i) $^nP_n = n \times {}^{n-1}P_{n-1}$

ii) $(n + 1) \times {}^nP_r = (n - r + 1) \times {}^{n+1}P_r$

iii) $^nP_r = {}^{n-1}P_r + r \times {}^{n-1}P_{r-1}$

iv) $1 \times {}^1P_1 + 2 \times {}^2P_2 + 3 \times {}^3P_3 + \dots + n \times {}^nP_n = {}^{n+1}P_{n+1}$

Solution: Permutation Identities

i) By definition:

$^nP_n = \frac{n!}{(n-n)!} = \frac{n!}{0!} = n!$ [∵ 0! = 1]

Again,

$n \times {}^{n-1}P_{n-1} = \frac{n(n-1)!}{[(n-1)-(n-1)]!}$

$= \frac{n(n-1)!}{0!}$

$= n \times (n-1)! = n!$

Hence, $^nP_n = n \times {}^{n-1}P_{n-1}$

ii) We have:

$(n+1) \times {}^nP_r = \frac{(n+1)n!!}{(n-r)!}$

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

Again,

$(n-r+1) \times {}^{n+1}P_r = \frac{(n-r+1)(n+1)!}{[(n+1)-r]!}$

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

Hence, $(n+1) \times {}^nP_r = (n-r+1) \times {}^{n+1}P_r$

Solution :

i) By definition

$$^np_n = \frac{n!}{(n-n)!} = \frac{n!}{0!} = n! \because 0! = 1]$$

Again $n \cdot {}^{n-1}p_{n-1} = n\frac{(n-1)!}{[(n-1)-(n-1)]}$

$$= n\frac{(n-1)!}{0!}$$

$$= n(n-1)! = n!$$

Hence, $^np_n = n. {}^{n-1}p_{n-1}$

ii) We have

$$(n+1)^n p_r = (n+1)\frac{n!}{(n-r)!} = \frac{(n+1)!}{(n-r)!}$$

Again $(n-r+1)^{n+1}p_r = (n-r+1) = \frac{(n+1)!}{[(n+1)-r]!}$

$$= (n-r+1)\frac{(n+1)!}{[(n+1)-r]!}$$

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

Hence, $(n+1)^n p_r = (n-r+1)^{n+1}p_r$

iii) We have,

$$^{n-1}\mathrm{p}_r + r\,^{n-1}\mathrm{p}_{r-1} = \frac{(n-1)!}{(n-1-r)!} + r\frac{(n-1)!}{[(n-1)-(r-1)]!}$$
$$= \frac{(n-1)!}{(n-1-r)!} + r\frac{(n-1)!}{(n-r)!}$$
$$= (n-1)!\left[\frac{1}{(n-r-1)!} + \frac{r}{(n-r)!}\right]$$
$$= (n-1)!\left[\frac{(n-r)}{(n-r)(n-r-1)!} + \frac{r}{(n-r)!}\right]$$
$$= (n-1)!\left[\frac{(n-r)}{(n-r)!} + \frac{r}{(n-r)!}\right]$$
$$= (n-1)!\left[\frac{n-r+r}{(n-r)!}\right]$$
$$= \frac{n(n-1)!}{(n-r)!}$$
$$= \frac{n!}{(n-r)!} = \,^{n}\mathrm{p}_r$$

iv) We have,

$$1 + 1.\,^{1}p_1 + 2.\,^{2}p_2 + 3.\,^{3}p_3 + \cdots \ldots \ldots + n.\,^{n}p_n$$
$$= 1 + 1.1! + 2.2! + 3.3! + \cdots \ldots \ldots + n.n!$$
$$= 1 + (2-1)! + (3-1)2! + (4-1)3! + \cdots \ldots \ldots [n+1-1]n!$$
$$= 1! + (2! - 1!) + (4! - 3!) + \cdots \ldots \ldots + [(n+1) - n!]$$
$$= (n+1)!$$
$$= \,^{n+1}p_{n+1}$$

Example 2:

How many three letter words with or without meaning can be formed out of the letters of the word SWING when repetition of letters is not allowed?

sol:-

here n = 5 (Swing has 5 letters)

we have to frame 3 letter words (r)

so permitation P(n,r) = $\frac{5!}{(5-3)!}$

$$= \frac{5*4*3*2*1}{2*1}$$

= 60

Example 3 :

How many 3 letter words with or without meaning can be formed out of letters of word SMOKE when repetition is allowed ?

sol:

SMOKE has 5 alphabets

so n = 5

we have to arrary in 3 form permutation (when repetition is allowed)

$5^3 = (5*5*5)$

= 125

Example 4:

It is reuired to seat 5 men and 4 women in a row so that the women occupy the even places.

how many such arrangements are possible ?

sol:

5 men and 4 women

i.e. total 9 positions

four places can be occupied by 4 women in p(4,4) ways = 4!

= 4*3*2*1

=24 ways

Remaining 5 positions can be occupied by 5 men i.e. 5 !

= 5*4*3*2*1

= 120 ways

Total no. of ways of seating arrangements = 24 * 120

= 2880 ways

Example 5:

Find the no. of words, with or without meaning, that can be formed with the letters of the words SWIMMING ?

sol:-

SWIMMING = 8 letters

here, I comes 2 times and M comes 2 times

no. of words formed

$$= \frac{8!}{2! * 2!}$$

$$= \frac{8 * 7 * 6 * 5 * 4 * 3 * 2 * 1}{(2 * 1) * (2 * 1)}$$

= 8*7*6*5*2*3

=10080

Example 6:

Find the number of words, that can be formed with lettes of the word INDIA

sol:-

INDIA = 5 words

'I' comes twice

Note:- when a letter comes more than once in a word, we divide the factorial of the no. of all letters in the words by the number of occurrences of each letter.

INDIA = $\frac{5!}{2!}$

$$= \frac{55 * 4 * 3 * 2 * 1}{2 * 1}$$

= 60

Example:

how many different words can be formed with the letters of the word 'Super' such that the vowels always come together ?

sol:-

SUPER = 5 letters

SPR (UE) i.e total 4 letters

S, P, R, UE

No. of way which U and E can be arranged in 2! = 2

we arrange 4 letter 4!

$4! = 4*3*2*1$

$= 24$

$=24*2 = 48$ ways

Example:

find the no. of different words that can be formed with the letters of word " BUTTER" so that the vowels are always together.

sol:

BUTTER contains 6 letters

U, E should always come together

so BTTR (UE)

so in total we have 5 words

i.e B, T, T, R, UE

i.e $\frac{5!}{2!}$

$= \frac{5*4*3*2*1}{2}$

= 60

No. of way U and E are arranged = 2!

total no. of permutations possible = 60*2

= 120 ways

Combinations:-

Each of the different selections or groups that can be made by taking some or all of them (irrespective of order) is called a combination.

Arrangement in a particular order

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

$^{n}C_{r}$ = No. of combinations

n = total no. of objects

r = no. of choosing objects

Example:

In a party, every person shakes hands with every other person. If there are 105 handshakes, find the no. of person in the party.

sol:-

let n be the no. of persons in the party

No. of hand shakes = 105

so, total no. of handshakes $^{n}C_{2}$

${}^{n}C_2 = 105$

$$\frac{n!}{2!(n-2)!} = 105$$

$$\frac{n*(n-1)}{2} = 105$$

$n^2 - n = 210$

$n^2 - n - 210 = 0$

n = 15, -14

we can not take -ve value

so n = 15

i.e. No. of persons in party = 15

Example:

A question paper has two part A and B each containing 10 questions. If a student has to choose 8 from part A and 5 from part B, in how many ways can he choose the questions ?

sol:-

Total questions = 10 in A and 10 in B

we have to choose 8 questions from A

$= {}^{10}C_8$

and 5 question from B

$= {}^{10}C_5$

hence total no. of ways

${}^{10}C_8 * {}^{10}C_5$

$$= \frac{10!}{8!(10-8)!} * \frac{10!}{5!(10-5)!}$$

$$= \frac{10!}{8!*2!} * \frac{10!}{5!*5!}$$

$$= \frac{10*9*8!}{8!*2*1} * \frac{10*9*8*7*6*5!}{5!*5*4*3*2*1}$$

= (5*9) * (2*9*2*7)

= 45*252

=11340

Probability

Total number of ways of achieving success or total number of possible outcomes is known as Probability.

$$P(A) = \frac{No. of\ favourable\ outcome}{Total\ no. of\ favourable\ outcomes}$$

P(A) = Probability of an event "A"

Example:

what is the probability that a card taken from a standard deck, is an ace ?

sol:

Total no. of cards = 52

No. of ace cards = 4

so, no. of favorable outcomes = 4

$$P(ACE) = \frac{No. of\ favourable\ outcomes}{Total\ no. of\ favorable\ outcomes}$$

$$= \frac{4}{52}$$

$$= \frac{1}{13}$$

Group Theory

GROUP :-

A monoid (G, ∗) with idntity e, is said to be a group if for every a ∈ G there exists an element b ∈ G such that a ∗b=e=b∗a.

b is known as inverse of a and we write a^{-1} = b.

Note that if b is an inverse of a, then a is an inverse of b.

(i) Associativity

(a ∗ b) ∗ c = a ∗ (b ∗ c), for any elements a, b and c in G.

(ii) Existence of Identity

There exists an element e ∈ G such that for all a∈ G

a ∗e = a = e ∗ a

e is called identity.

(iii) Existence of Inverse

For every a ∈ G, there exists b ∈ G such that

a ∗ b = e = b ∗ a

b is called inverse of a.

observe that if (G, ∗) is a group, then ∗ is a binary operation on G, so G must be closed under *, that is a ∗ b ∈ G for any elements a and b in G.

Furthermore, if ∗ is commutative, G is called an Abelian group.

Abelian Group:-

A group is abelian if the operation is commutative, i.e.

a ∗ b = b ∗ a

for all a, b ∈ G

A group (G,*) is nonabelian or non commutative, if there is some pair of elements a and b in G for which a ∗ b ≠ b ∗ a

Type of Groups:

finite Group:- A group with a finite number of elements. The number of elements in a group is called the order of the group.

if G is a group that has a finite number of elements, we say that G is a finite group, otherwise the group is infinite.

Infinite Group :- A group with an infinite number of elements.

Cyclic Group :-

A group generated by a single element. if a ∈ G generates all elements, G is cyclic, and a is a generator of G.

such that every element of G can be written in the form of a^n for same n ∈ Z.

The additive group Z of integers is a cyclic group generated of 1, since 1 ∈ Z and for every integer n, we have n = n.1

we see that -1 is also a generator of Z, since n = (-n)(-1) for every n ∈Z.

thus Z =< 1 > = <-1>

Order of an Element :- The smallest positive integer n such that a^n = e, is called the order of a.

where e is the identity.

if no such n exists, the elements has infinite order.

clearly the identity element of a group is the only element of order one.

Note that if 0(a) = n and if for some positive integer m,

a^m = e them m is multiple of n.

Sub Group:-

A non-empty subset H of a group G is a subgroup of G if H is also a group under the same binary operation of G. Symbolically, we write H ≤ G.

If e is the identity of G, {e} is also a subgroup of G, called the trivial subgroup of G. All other subgroups are nontrivial.

G itself is a subgroup of G, called the improper subgroup of G.

All other subgroups are proper.

Graph Theory

Graph are discrete structures consisting of vertices and edges that connects these vertices. There are several different types of graphs that differ with respect to the kind and number of edges that can a connect a pair of vertices. Problems in almost every conceivable discipline can be solved using graph models.

Basic terminology :-

A **graph** is a collection of **nodes (called vertices)** and **connections (called edges)** between them.

A graph is written as: **G = (V, E)**

Where

- **V** = set of vertices
- **E** = set of edges (pairs of vertices)

Vertex (Node) – A point in the graph

Edge (Link) – A line connecting two vertices

Adjacent – Two vertices are adjacent if they are connected by an edge if there is an edge e = (u, v) connecting vertices u and v, then u and v are called adjacent to each other.

Degree of Vertex – Number of edges connected to a vertex

Indegree: Number of incoming edges (for directed graphs)

Outdegree: Number of outgoing edges

Even and odd vertices : A vertex is said to be an even or odd vertices according as its degree is an even or odd number.

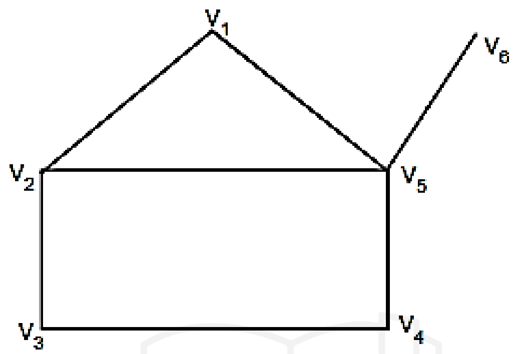
Path – A sequence of edges connecting a sequence of vertices

Cycle – A path that starts and ends at the same vertex

The most common representation of a graph is by means of a diagram in which the vertices are represented as points and each edge as a line segment joining its end vertices.

Graph Types :-

1. **undirected graph :-** A graph is said to be an undirected graph if edges are unordered pairs of distinct vertices.



   In an undirected graph we can refer to an arc joining the vertex pair u and v as either (u, v) or (v, u).

2. **directed graph :-** A graph is said to be the directed graph if the edges are ordered pairs of vertices. In this case an edge (u, v) is said to be from u to v and to join u to v.

3. **Weighted Graph :-**
   - Every edge has a **weight or cost**.
   - Weight can represent **distance, time, cost, bandwidth**, etc.
   - Can be either **directed** or **undirected**.

4. **Unweighted Graph :-**
   - No weights are assigned to edges.
   - All connections are treated **equally**.

5. **Simple Graph:-**
   - A graph with **no loops** (edge from a node to itself)
   - And no multiple edges between same nodes

6. **Multigraph :-**
   - Allows **multiple edges** between two vertices
   - Can be **directed or undirected**

7. **Complete Graph (K_n) :-**
   - Every pair of vertices is **connected by exactly one edge**
   - A complete graph on n vertices may be denoted by the symbol k_n.
   - The degree of every vertex is n–1 in a complete graph of n vertices.
   - Number of edges = **n(n – 1)/2**

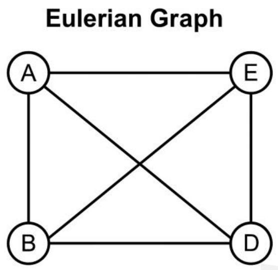
8. **Cyclic Graph:-**
   - A graph that has **at least one closed loop or cycle**
   - Cycle: A → B → C → A

9. **Connected Graph (Undirected) :-**
   - Every vertex is **reachable from every other vertex**
   - Only one component
10. **Disconnected Graph:-**
    - Has **isolated vertices** or groups (components) not connected to others
11. **Regular Graph:-** A graph in which every vertex has the same degree is called a regular graph. If every vertex has degree K, then the graph is called a k-regular or regular graph of degree K.

Eulerian graph

A **graph is Eulerian** if it contains a **Eulerian circuit**, which means:

A **closed trail** (path) that visits **every edge exactly once** and returns to the **starting vertex**.



Rules:-

1. Have even degree in all vertics. Allow odd degree only in two vertices.
2. Edge repeat only once vertices multiple times repeated.
3. Always connected i.e. always connect all the eges.

   A path that passes through each edge exactly once but vertices may be repeated is called Euler path.

   A graph that contains an Euler tour or Euler circuit is called an Eulerian graph.

1. **Eulerian Trail (or Path)**
   - A **trail** that passes through **every edge exactly once**.
   - It **does not have to be a cycle**, so it may start and end at different vertices.
2. **Eulerian Circuit (or Cycle)**
   - A **closed trail** that starts and ends at the **same vertex**.
   - It **uses every edge exactly once**.
   - Graph is **connected** and **all vertices have even degree.**

Hamiltonian Graphs

A **Hamiltonian graph** contains a **Hamiltonian cycle**, which means:

A **closed loop** that visits **every vertex exactly once**, and returns to the **starting vertex**.

Rules:

1. Travel vertices only once
2. First and last vertex are same .
3. This is rule to check graph is hamilitonian or not.
4. No need to travel all edges.