Chapter: 1

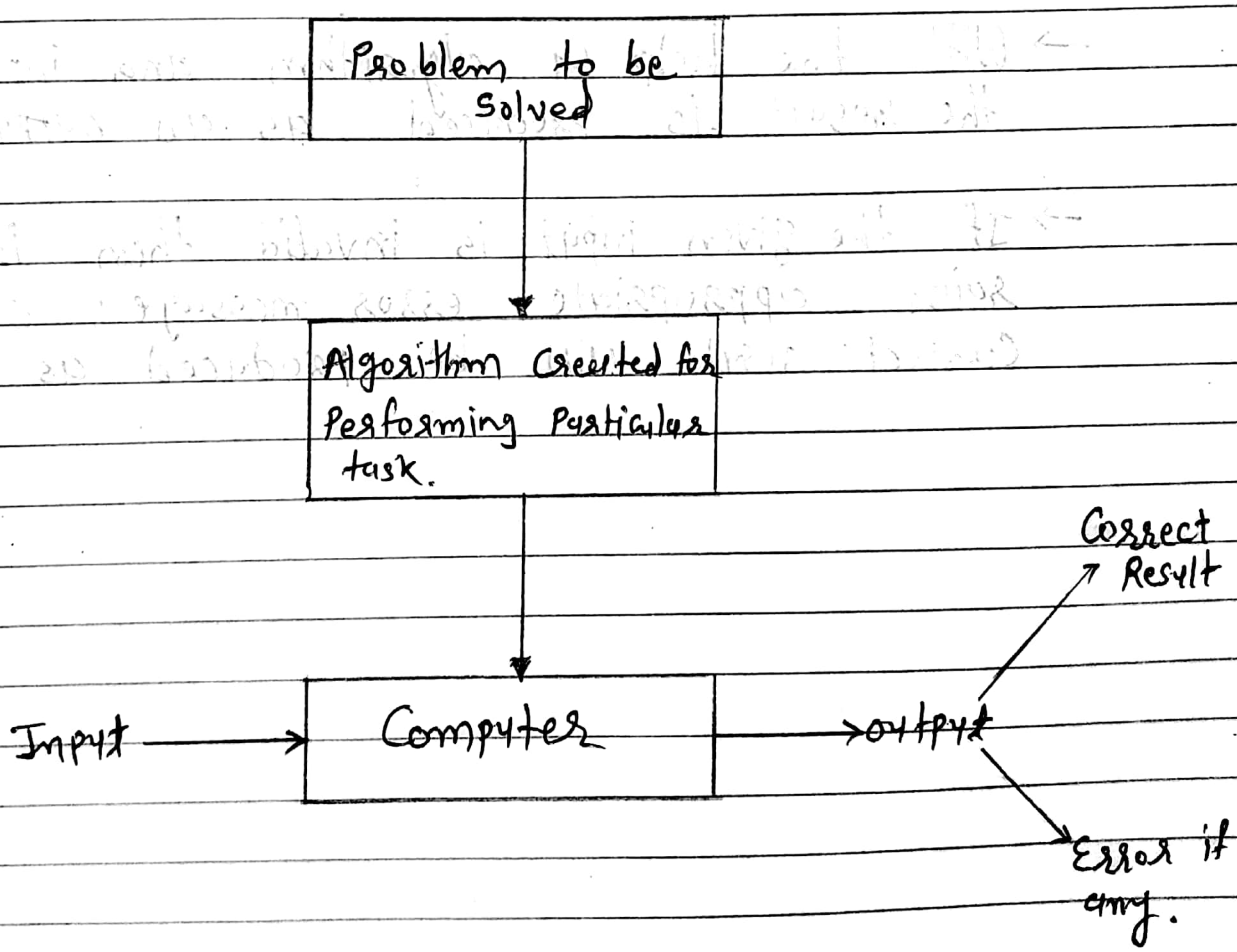
Basics of Algorithms and Mathematics.

⊛ What is an Algorithm?

"The Algorithm is defined as a Collection of unambiguous instructions occurring in some specific sequence and such an algorithm should produce output for given set of input in finite amount of time."

one meaning

→ This definition of algorithm is represented in fig.



[Notion of algorithm]

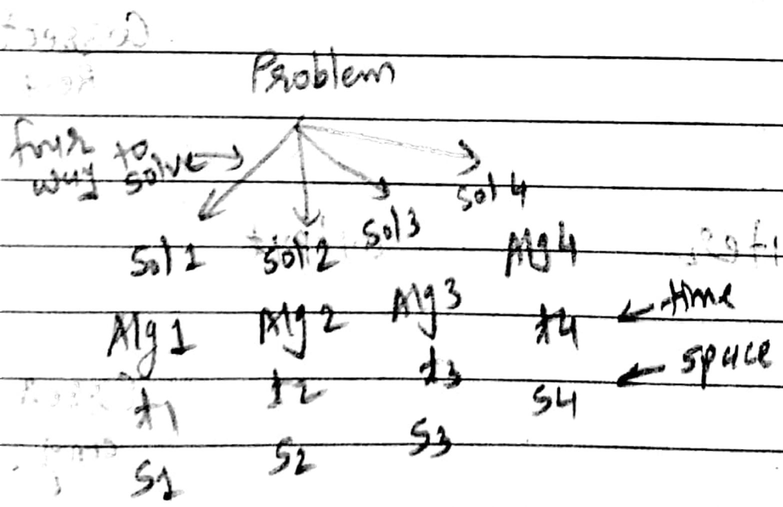
→ After understanding the problem statement we have to create an algorithm carefully for the given problem.

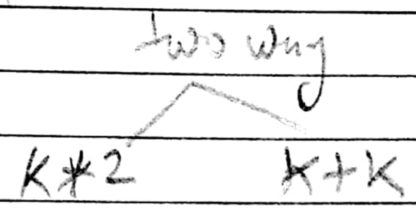
→ The algorithm is then converted into some programming language and then given to some computing device (computer).

→ The computer then executes this algorithm, during the process of execution it requires certain set of input.

→ With the help of algorithm and input set, the result is produced as an output.

→ If the given input is invalid then it should raise appropriate error message; otherwise correct result will be produced as an output.





⇒ Properties of Algorithm :-
OR
Characteristics of Algorithm

→ It is necessary to have following properties associated with an algorithm :-

1) Non-ambiguity :- give same output for every time
3+4 = 7

Each Step in an algorithm should be non-ambiguous. That means each instruction should be clear and precise. (accurate)

- The instruction in an algorithm should not denote any Conflicting meaning. This property also indicate the effectiveness of algorithm.

2) Range of input :-

The Range of input should be specified. This is because normally the algorithm is input driven and if the range of the input is not been Specified then algorithm Can go in an infinite state.

3) Multiplicity :-

The Same algorithm Can be represented in several different ways.

- That means We can Write in simple English the Sequence of instructions or we can write it in the form of pseudo Code.

→ for Solving the same problem we can write ~~it in~~ ~~#~~ several different algorithms.

– for instance:- for searching a number from the given list we can use Sequential Search or a binary search method.

4) Speed:-
The algorithms are written using some specific ideas. But such algorithms should be efficient and should produce the output with fast speed.

5) Finiteness:-
The algorithm should be finite. That means after performing required operations it should terminate.

⇒ Issues in Writing an algorithm:

→ There are various issues in the study of algorithm and those are:-

1) How to devise algorithms?
(Creation)

→ The creation of an algorithm is a logical activity and one cannot automate it. But there are certain algorithmic design strategies and using these strategies

one can create many useful algorithms.
- Hence Mastering of Such design strategies is an Important activity in study of design and analysis of algorithms.

2) How to Validate algorithms ?

→ The Next step after Creation of algorithm is to Validate algorithms.

- The process of checking whether an algorithm Computes the Correct answer for all possible legal inputs is called algorithm Validation.

- The purpose of Validation of algorithm is to find whether algorithm works properly without being dependent upon programming language.

3) How to Analyze algorithms ?

- Analysis of algorithm is a task of determining how much Computing time and storage is required by an algorithm. ✓

- Analysis of algorithm is also Called performance analysis.

- This analysis is based on mathematics and a judgment is often needed about better algorithm when two algorithm get Compared.

- The Behaviour of algorithm in best case, worst case and average case needs to be obtain.

4) How to test a Program?

- After finding an efficient algorithm it is necessary to test that the program written using the efficient algorithm behaves properly or not.

- Testing of a program is an activity that can be carried out in two phases:

① Debugging :-
Debugging is checked whether program produces faulty results for valid set of input and if it is found then the program has to be corrected.

② Performance measuring :-
Performance measuring is a process of measuring time and space required by a corrected program for valid set of inputs.

⇒ How to Write an Algorithm?

→ Algorithm is basically a sequence of instructions written in simple English language. The Algorithm is broadly divided into two Section.

Algorithm Heading
It Consists of name of algorithm, Problem description, input & output.

Algorithm Body
It Consists of logical body of the algorithm by making use of various programming Constructs & Assignment Statement.

↳ Let us Understand Some rules for Writing the algorithm.

1) Algorithm is a Procedure Consisting of heading and body. The heading Consists of Keyword Algorithm and Name of the algorithm and parameter list. The Syntax is.

Algorithm name(P1, P2, ... Pn)

This keyword should be written first.

Here write the name of the Algorithm

Write Parameters.

2) Then in the heading section we should write following things:

// Problem Description:
// Input:
// output:

3) Then body of an algorithm is written, in which various programming constructs like if, for, while or some assignment statements may be written.

4) Compound statements should be enclosed within { and } brackets.

5) Single line comments are written using // as beginning of comment.

6) The identifier should begin by letter and not by digit. An Identifier can be a combination of alphanumeric string

7) Using assignment operator ← an assignment statement can be given.

$$\boxed{\text{Variable} \leftarrow \text{expression.}}$$

8) These are other types of operators such as Boolean operators such as true or false. Logical operators such as and, or, not. and Relational operators such as <, <=, >, >=, =, ≠.

9) The Array indices are stored with in square brackets '[' ']'. The Index of array usually start at zero.

10) The Inputting and outputting can be done using read and write

Ex:-

Write ("This message will be displayed on Console");

read (Val);

11) The Conditional Statements such as if-then or if-then-else are written in following form :-

if (condition) then statement

if (condition) then statement else statement.

12) While statement can be written as:-

```
while (condition) do
{
    statement 1
    statement 2
    :
    statement n
}
```

- While the condition is true the block enclosed with { } gets executed otherwise statement after } will be executed.

13) The general form for writing for loop is:-

for Variable ← $Value_1$ to $Value_n$ do
{
Statement 1
Statement 2
⋮
Statement n
}

↳ Here $Value_1$ is initialization condition and $Value_n$ is a terminating condition.

14) The repeat - until statement can be written as:

repeat
Statement 1
Statement 2
⋮
Statement n
until (condition)

15) The break statement is used to exit from inner loop.

⇒ Example 1: Write an Algorithm to ~~Count~~ find the sum of n numbers.

↳ Algorithm sum (1, n)
//Problem Description: This algorithm is for finding
// the sum of given n number
// input: 1 to n numbers.
// output: The sum of n numbers.

~~sum result ← 0~~

sum ← 0
for i ← 1 to n do
sum ← sum + i

return sum.

⇒ Example 2: Write an algorithm to check whether given number is even or odd.

↳ Algorithm eventest (val)
// Problem Description: This algorithm test whether
// given number is even or odd
// Input: the number to be tested i.e. val
// output: Appropriate message indicating even or oddness

if (val % 2 = 0) then
Write ("Given number is even")
else
Write ("Given number is odd")

Scanned by CamScanner

⟹ **Example 3:** Write an algorithm to find factorial of n number.

→ Algorithm fact(n)
//Problem Description: This algorithm finds the
// factorial of given number n.
// Input: The number n of which the
// factorial is to be calculated
// Output: factorial value of given n.

if (n <= 1) then
return 1
else
return n * fact(n-1).

Scanned by CamScanner

⊛ Designing of an Algorithm:

↳ Before Actual implementation of the program, designing an algorithm is very important step.

↳ Suppose, if we want to build a house we do not directly start constructing the house.

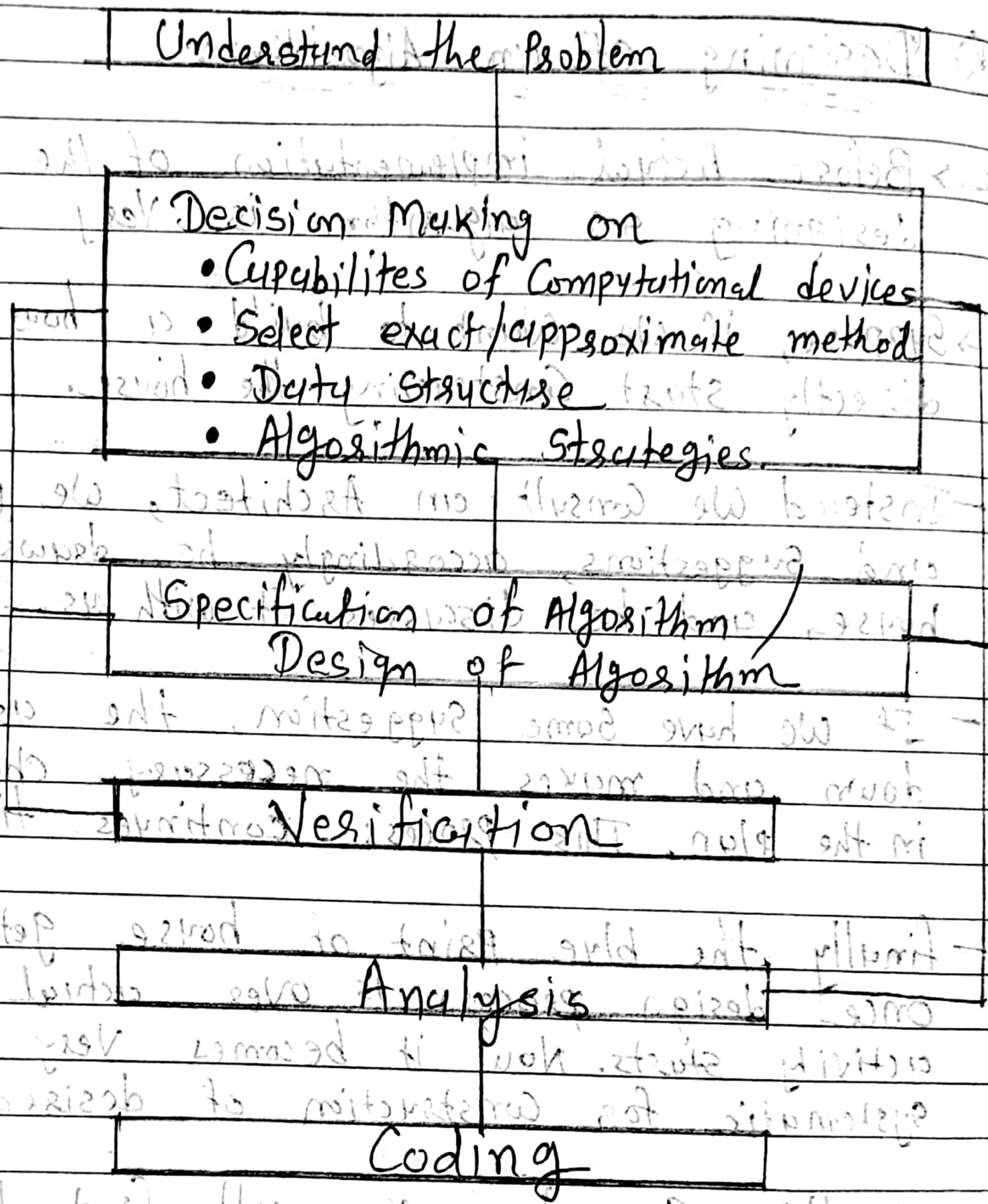
- Instead we consult an Architect, we put our ideas and suggestions, accordingly he draws a plan of the house, and he discusses it with us.

- If we have some suggestion, the architect notes it down and makes the necessary changes accordingly in the plan. This process continues till we are happy.

- finally the blue print of house gets ready. once design process is over actual construction activity starts. Now it becomes very easy and systematic for construction of desired house.

- In this Example, you will find that all designing is Just a paper work and at that instance if we want some changes to be done then those can be easily carried out on the paper.

→ Let us list the "What are the steps that need to be followed?" while designing an algorithm.



[Algorithm design steps]

↳ Let us now discuss each step in detail.

1) Understanding the Problem

- This is the very first step in designing of algorithm. In this step first of all you need to understand the problem statement completely.

- While understanding the problem statements, read the problem description carefully and ask questions for clarifying the doubts about the problem.

- After carefully understanding the problem statements find out what are the necessary inputs for solving that problem. The input to the algorithm is called instance of the problem.

- It is very important to decide the range of inputs so that the boundary values of algorithm get fixed.

2) Decision Making

- After finding the required input set for the given ~~algorithm~~ problem we have to analy the input and need to decide certain issues such as capabilities of computational devices, whether to use exact or approximate problem solving, which data structures has to be used and to find the algorithmic technique for solving the given problem.

(a) Capability of Computational devices:-

- It is necessary to know the computational capabilities of devices on which the algorithm will be running.

- Globally we can classify an algorithm from execution point of view as sequential algorithm and parallel algorithm.

  - There are certain complex problems which require huge amount of memory or the problems for which execution time is an important factor. For solving such problems it is essential to have proper choice of a computational device which is space and time efficient.

(b) Choice for either exact or approximate problem solving method.

- The next decision is to decide whether the problem is to be solved exactly or approximately

- If the problem needs to be solved correctly then we need exact algorithm. otherwise if the problem is so complex that we won't get the exact solution then in that situation we need to choose approximation algorithm. Ex- Travelling Salesman problem.

(c) Data Structures

- Data structure and algorithm work together and these are interdependent.

- Hence choice of proper data structure is required before designing the actual algorithm

- The Implementation of algorithm (Program) is possible with the help of algorithm and data structure.

(d) Algorithmic Strategies:-

- Algorithm Design Techniques:-

- **Brute force**:- This is straightforward technique with naive approach.

- **Divide and Conquer**:- The problem is divided into smaller instances.

Scanned by CamScanner

• **Dynamic Programming** :- The result of smaller reoccurring instances are obtained to solve the problem.

• **Greedy technique** :- ~~This~~ ~~meth~~ To solve the problem locally optimal decisions are made.

• **Back tracking** :- This method is based on the trial and error. If we want to solve some problem then desired solution is chosen from the finite set S.

3) Specification of Algorithm

→ There are various ways by which we can specify an algorithm.

Algorithm →
- Using Natural language
- Pseudo code
- flowchart.

→ for Example : Write an algorithm to perform addition of two numbers.
(Natural language)

Step 1 : Read the first number say a.
Step 2 : Read the second number say b.
Step 3 : Add the two numbers and store the result in c.
Step 4 : Display the result.

Scanned by CamScanner

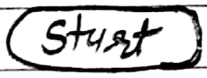
Using Pseudo Code

→ For Example: Write an algorithm for performing addition of two numbers.

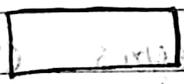
Algorithm Sum(a, b)
// Problem Description: This algorithm performs addition of
// two numbers.
// Input : Two integers a and b.
// output : addition of two integers.

c ← a+b
Write (c).

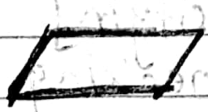
→ Another way of representing the algorithm is by flowchart. Flowchart is a graphical representation of an ~~flow~~ algorithm. Typical symbols used in flowchart are :-



Start State



Transition

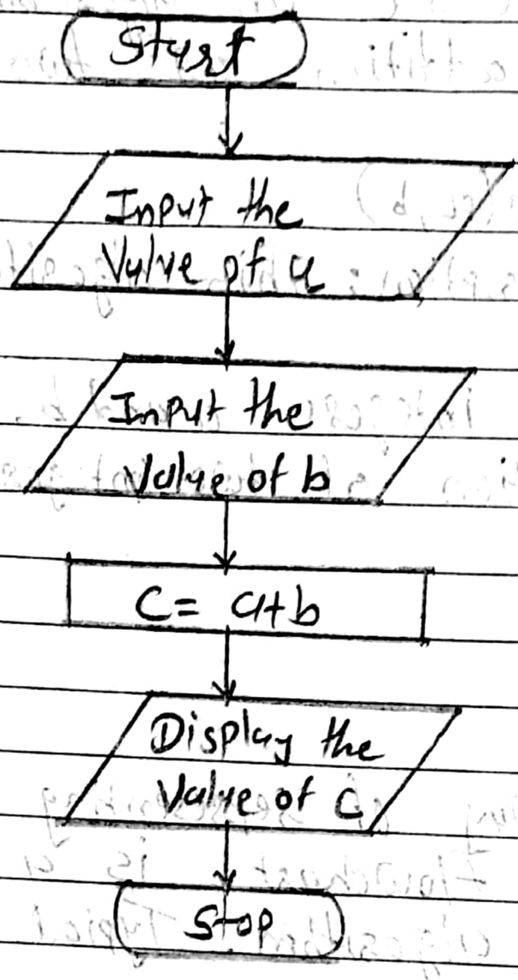


Assignment Statements / Input-output.

Conditional Statement

Stop. State

→ for Example



4) Verification of Algorithm.

→ Algorithmic Verification means checking of an algorithm. After Specifying an algorithm we go for checking its Correctness.

→ We normally check whether the algorithm gives Correct output in finite amount of time for a Valid Set of input.

5) Analysis of Algorithm :-

→ While analyzing an algorithm we shold consider following factors :-

- **Time Complexity :-**
  Time Complexity of an algorithm means the amount of time taken by an algorithm to run.

- **Space Complexity :-**
  Space Complexity of an algorithm means the amount of space (memory) taken by an algorithm. We can analyze whether an algorithm requires more or less space.

- **Simplicity :-**
  Simplicity is of an algorithm means generating sequence of instructions which are easy to understand.
  - simple algorithm can be understood quickly & one can then write simpler programs for such algorithm.
  - finding out bugs from algorithm or debugging the progrum becomes easy when an algorithm is simple.

- Generality :-
  Generality Sometimes it becomes easier to design an algorithm in more general way rather than designing it for particular set of input.

- Hence we should write general algorithm always.
- for example, designing an algorithm for finding GCD of any two numbers is more appealing than that of particular two numbers.

- Range of Inputs :-
  Range of inputs Comes in picture when we execute an algorithm The design of an algorithm should be such that it should handle the range of input which is the most natural to Corresponding Problem.

6) Implementation of algorithm

- The Implementation of an algorithm is done by suitable programming language.

⊛ Mathematics for Algorithmic Set

→ "Set is defined as Collection of objects."

→ These objects are called elements of the set. all the elements are enclosed within curly brackets '{' and '}' and every elements is separated by commas.

↳ Set is denoted by a Capital letter.

↳ The set can be represented using two methods

① Listing method

↳ the elements are listed in the set.

$A = \{0, 1, 2, 3, 4\}$

$B = \{1, 3, 5, 7, 9\}$

② By Properties.

$A = \{x / x$ is set of elements which are less than 5 $\}$

$B = \{x / x$ is odd number which are less than 10 $\}$

→ **Subset :**

The Subset A is Called Subset of Set B If every element of set A is Present in Set B but reverse is not true.

- It is denoted by $A \subseteq B$.

Ex: $A = \{1, 2, 3\}$ &

$B = \{1, 2, 3, 4, 5\}$ then $A \subseteq B$.

→ **Empty Set :**

The set having no elements in it is called empty set. it is denoted by $A = \{\ \}$ or ϕ

→ **Null String :-**

The null element is denoted by ε or Λ character. Null element means no value character.

→ **Power Set :-**

The power set is a set of all the subsets of its elements.

Ex: $A = \{1, 2, 3\}$

Power set : $Q = \{\phi, \{1\}, \{2\}, \{3\}, \{1,2\}, \{1,3\}, \{2,3\}, \{1,2,3\}\}$

No. of elements $= 2^n = 2^3 = \underline{8}$

→ **Equal Set :-**

The two sets are said to be equal $(A=B)$ if $A \subseteq B$ and $B \subseteq A$.
i.e. every element of set A is an element of B and every element of B is an element of A.

Ex: $A = \{1,2,3\}$ & $B = \{1,2,3\}$ then $A=B$

Operations on Set :-

→ Various operations that can be carried out on set are-

1. Union
2. Intersection
3. Difference
4. Complement.

→ ① **Union Operation**

$A \cup B$

$A = \{1,2,3\}$ & $B = \{1,2,4\}$ then

$A \cup B = \{1,2,3,4\}$ Combination of both set.

→ ② **Intersection Operation** $(A \cap B)$

$A = \{1,2,3\}$ & $B = \{2,3,4\}$

$A \cap B = \{2,3\}$, Collection of ~~Common element~~ from both the sets.

→ (3) **Difference** $(A-B)$

$A = \{1,2,3\}$ & $B = \{2,3,4\}$ then

$A-B = \{1,2,3\} - \{2,3,4\}$

$= \{1\}$

i.e. - elements which are there in set A but not in set B.

→ (4) **Complement** $(\bar{A})$

$A' = U - A.$

$U = \{1,2,3,4,5\}$

$A = \{2,4\}$

$\bar{A} = A' = \{1,3,5\}.$

⟹ **Cartesian Product of two Sets :-**

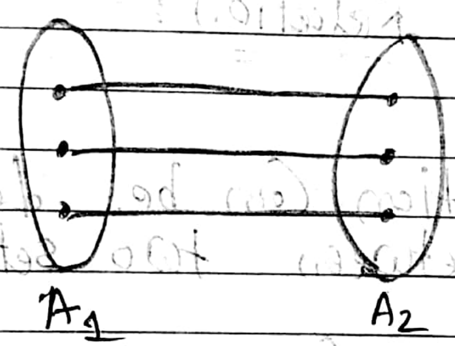
- The Cartesian product of two sets A and B is a set of all possible ordered pairs whose first element (component) is member of A and whose second component is member of B. The Cartesian product is denoted by $A \times B$.

$A = \{a, b\}$ $B = \{0, 1, 2\}$

$A \times B = \{(a,0), (a,1), (a,2), (b,0), (b,1), (b,2)\}$

⇒ **Cardinality of Sets :-**

↳ Cardinality of Set is nothing but the numbers of members in the set.



- These sets A_1 and A_2 have the same Cardinality as there is one to one mapping of the elements of A_1 & A_2.
- The different Cardinalities of set can be one to one, one to many, many to one and many to many.

⇒ **Sequence :-** → Same type of elements (like Array)

The Sequence means Ordering of the elements of the Set in some specific manner. for instance: {0, 2, 4, 6, 8} is a Sequence which denotes the even numbers.

Ex :- {0, 1, 1, 2, 3, 5, 8, 13} ← Fibonacci Sequence

Maybe Different types elements.

⇒ **Tuple :-**

An Ordered pair of n elements is called tuple.

{10, 20} is a 2-tuple of pair
{10, 20, 30} is 3-tuple.

⊛ Function and Relation :-

⇒ **Function :-**

function can be defined as the relationship between two sets.

- That means using function we can map one element of one set to some other element of another set.

- A function can be denoted using a letter f.

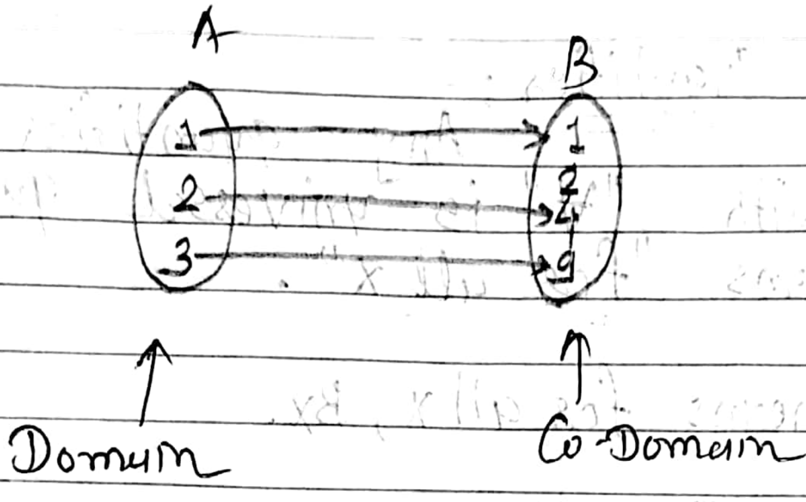
If $f(x) = x^2$ then we say that f of x equals to x square.

$f(1) = 1$

$f(2) = 4$

$f(3) = 9$

and so on.



Domain = $\{1, 2, 3\}$

Co-Domain = $\{1, 2, 4, 9\}$

Range = $\{1, 4, 9\}$

Pair = $(1,1), (2,4), (3,9)$

→ **Example :-** find the domain and range of the following relation. Is this relation a function $\{(2,9), (3,14), (4,21)\}$

→ Domain = $\{2, 3, 4\}$

Range = $\{9, 14, 21\}$

⇒ **Quantifiers :-** In Predicate logic the quantifier is a kind of operator which is used to determine the quantity.

→ Two types of quantifiers :- ① Universal
② Extential

→ **Universal quantifier:-** Any quantifier that starts with "$\forall$" is universal quantifier. $\forall x$ means "for all x".

Ex: $\forall x Bx$ means for all x, Bx.

→ **Extential Quantifier:-** Any quantifier that starts with "$\exists$" is an extential quantifier

$\exists x$ means there exists an x such that...

Ex: "Some cars use red in Color"

⇒ **Relations:-** "The relation R is a Collection for the set S which represents the pair of elements."

→ One object can be related with the other object by a mother of relation. Then those two objects from a pair based on this certain relationship.

→ Ex: (a,b) is in R. we can represent their relation as aRb.

- The first component of each pair is chosen from a set called domain and second

Component of each pair is chosen from a set called range.

⇒ Properties of Relations:-

A Relation R on sets S is,

1. Reflexive if iRi for all i in S. {(a,a),(b,b)}
2. Irreflexive if iRi is false for all i in S. {(a,b)}
3. Transitive if iRj and jRk then iRk. {(a,b),(b,c),(a,c)}
4. Symmetric if iRj implies jRi {(a,b),(b,a)}
5. Asymmetric if iRj implies that jRi is false. {(a,b)}

⇒ Equivalent Relation:-

"A relation is said to be equivalence relation if it is reflexive, symmetric and transitive over some set S."

Ex:- Determine whether R is equivalence relation or not where A={0,1,2}, R={(0,0),(1,0),(1,1),(2,2),(2,1)}.

↳ The R is reflexive because, (0,0),(1,1),(2,2) ∈ R

↳ Where as R is not symmetric because (0,1) is not in R whereas (1,0) is in it.

Hence R is not a equivalence relation.

⇒ Closures of Relations:-

Sometimes when the relation R is given, it may not be reflexive or transitive. By adding some pairs we make the relation as reflexive or transitive.

Ex: Let {(a,b), (b,c), (a,a), (b,b)}.

Transitive Closures = {(a,a), (b,b), (a,b), (b,c), (a,c)}.

Symmetric closures = {(a,b) (b,a), (b,c), (c,b) (a,a) (b,b)}

Reflexive closures = {(a,a), (b,b), (c,c)}.

⊛ Vectors :-

"A Vector A is a Collection of n tuples."

Ex: $A = \{a_1, a_2, a_3, ---, a_n)$

a_1 are called the Components of A.

→ Equal Vectors:

If there exists two Vectors namely, A and B and if both the vector contain same number of Components and if Corresponding Components are equal then Vector A=B. i.e. Vector A and B are equal.

Scanned by CamScanner

→ Zero Vector :-

Let $A = (a_1, a_2, a_3 \dots a_n)$ be a Vector and if $a_i = 0$ then Vector A is called Zero Vector.

→ Addition of two Vectors :-

For addition of two Vectors, the numbers of Components in both the Vectors must be the same. The Sum A+B is a Vector obtained by adding Corresponding Components from A and B.

$$A + B = (a_1, a_2, a_3 \dots, a_n) + (b_1, b_2, b_3, \dots b_n)$$

$$= (a_1 + b_1, a_2 + b_2, a_3 + b_3, \dots a_n + b_n)$$

→ Multiplication of Vector by scalar :-

- By multiplying each Component of Vector by a Scalar the product is obtained.

- Let K be a Scalar then the product can be.

$$K \cdot A = K \cdot (a_1, a_2, \dots, a_n)$$

$$= K \cdot a_1, K \cdot a_2, \dots K a_n$$

→ **Dot Product :-** The dot product or inner product of Vectors $A = (a_1, a_2, a_3, \ldots, a_n)$ and $B = (b_1, b_2, b_3, \ldots, b_n)$ is denoted by $A \cdot B$.

It can be defined as follows,

$$A \cdot B = a_1 b_1 + a_2 b_2 + a_3 b_3 + \cdots + a_n b_n.$$

→ **Length of a Vector :-** A length or norm of the Vector, A is denoted by $\|A\|$. & It

$$\|A\| = \frac{\sqrt{A \cdot A}}{\sqrt{a_1^2 + a_2^2 + \cdots + a_n^2}}$$

⊛ Matrices :-

In Mathematics, an array is rectangular representation of numbers.

$$A = \begin{bmatrix} 11 & 12 & 13 \\ 8 & 7 & 5 \\ 15 & 20 & 25 \end{bmatrix}$$

- Horizontal lines in a matrix are called rows and Vertical lines are called Columns.
- The size of the matrix can be specified with m rows and n Columns. It is denoted as $m \times n$ or m by n where m and n are called its dimensions.

→ Row Vector and Column Vector

A matrix with one row is called row vector and matrix with one Column is called Column Vector.

$$\begin{bmatrix} 10 & 20 & 30 \end{bmatrix}$$

Row Vector

$$\begin{bmatrix} 11 \\ 12 \\ 13 \end{bmatrix}$$

Column Vector

① Zero Matrix :-
A Zero Matrix is a matrix with all its entries being Zero.

Ex.- $O_{2,2} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$

② Identity Matrix :-
The Identity matrix or unit matrix of size n is a square matrix having One's on the main diagonal and Zero's elsewhere.

- The Identity matrix is denoted by I.

Ex:-

$I_1 = [1]$

$I_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$

$I_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

③ Square Matrix :-
If the number of rows & number of Columns of any matrix are same then we say that the matrix is square matrix.

⇒ Operations on Matrix :-

- Various operations that can be performed on matrix are :-

(1) Addition (2) Multiplication (3) Transpose

(1) Addition of two Matrices :-

- Let A and B are the two matrices of same size. Then the addition of these two matrices will be addition of each corresponding element.

Ex:

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$

$$B = \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{bmatrix}$$

$$A+B = \begin{bmatrix} a_{11}+b_{11} & a_{12}+b_{12} & a_{13}+b_{13} \\ a_{21}+b_{21} & a_{22}+b_{22} & a_{23}+b_{23} \\ a_{31}+b_{31} & a_{32}+b_{32} & a_{33}+b_{33} \end{bmatrix}$$

② Multiplication of two Matrices :-

- for multiplication of two matrices, width of first matrix equals to height of second matrix.

- That means a matrix A with $m \times n$ can be multiplied with matrix B having $n \times p$ size. Then the size of resultant matrix is $m \times p$.

Ex :-

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}_{3\times 2} \qquad B = \begin{bmatrix} 10 & 11 & 12 \\ 13 & 14 & 15 \end{bmatrix}_{2\times 3}$$

$$A \times B = \begin{bmatrix} (1*10)+(2*13) & (1*11)+(2*14) & (1\times 12)+(2\times 15) \\ (3\times 10)+(4\times 13) & (3\times 11)+(4\times 14) & (3\times 12)+(4\times 15) \\ (5\times 10)+(6\times 13) & (5\times 11)+(6\times 14) & (5\times 12)+(6\times 15) \end{bmatrix}$$

$$A \times B = \begin{bmatrix} 36 & 39 & 42 \\ 82 & 89 & 96 \\ 128 & 139 & 150 \end{bmatrix}_{3\times 3}$$

(3) Transpose of Matrix :-

- The transpose of matrix A is obtained by interchanging row and Column.
- The Transposed matrix is denoted by A^T.
- If Matrix A is of size $m \times n$ then A^T is $n \times m$

$$A = \begin{bmatrix} 10 & 20 \\ 30 & 40 \\ 50 & 60 \end{bmatrix} \text{ then } A^T = \begin{bmatrix} 10 & 30 & 50 \\ 20 & 40 & 60 \end{bmatrix}$$

⇒ Determinant of Matrix :-

- The determinant of Matrix A is a Specific number. It is denoted by $|A|$.

→ The determinant of Order one matrix is

$$|a_{11}| = a_{11}$$

→ The determinant of order two matrix is

$$\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} = a_{11}a_{22} - a_{12}a_{21}$$

→ The determinant of order three matrix is

$$\begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix}$$

$$= a_{11}(a_{22}a_{33} - a_{23}a_{32}) - a_{12}(a_{21}a_{33} - a_{31}a$$

$$+ a_{13}(a_{21}a_{32} - a_{31}a_{22}).$$

Linear Inequalities :-

In mathematics, Linear inequality is a statement Containing <, >, <=, >=.

⇒ Solution to Linear Inequality :-

→ The Solution of linear inequality is based on number of Variables.

① The Solution to one Variable inequality.

→ $3x+7 \leq 10$, the Value of x gives true Statement.

- For instance : $x=1$ is a solution to the above given inequality

② The Solution to two Variable Inequality

→ $3x+7y-5 \leq 17$, the Value of x and y Should be such that the Statement remains true.

- The Solution to above inequality is (2,2) because $3(2)+7(2)-5 = 15 \leq 17$.

③ The Solution to three Variable inequality

- $3x+2y-7z \leq 5$, the Value of x, y and z should be such that the statement remains true.
- the Solution to above inequality is (2,2,1).

Properties of Inequalities:

① If $x \le y$ & $y \le z$ then $x \le z$.

② If $x \le y$ & c is some positive number the $x + c \le y + c$.

③ If $x \le y$ & c is negative then $x \cdot c \ge y \cdot c$

Linear Equations :-

The linear equation is an equation containing n unknowns. A linear equation with one unknown can be given in ~~Statement~~ standard form:

$$ax = b$$

- where x is unknown and a & b are constants. The solution of such equation

$$x = b/a.$$

→ The linear equation with two unknowns can be given in standard form as

$$ax + by = c.$$

where x, y are unknowns and a, b & c are constants.

⇒ Two linear equations with two unknowns.

- The two linear equations with two unknowns is in following standard form:

$$a_1x + b_1y = c_1$$
$$a_2x + b_2y = c_2$$

Ex:

$$5x + 10y = 15$$
$$3x + 2y = 5$$

eqⁿ ① multiply by 3.
eqⁿ ② multiply by -5.

$$\cancel{15x} + 30y = 45$$
$$-\cancel{15x} - 10y = -25$$

$$20y = 20$$

$$\boxed{y = 1}$$

Put $y = 1$ in eqⁿ ②

$$3(x) + 2(1) = 5$$

$$3x = 3$$

$$\boxed{x = 1}$$

Gauss Elimination Method

→ Gauss elimination method is a method of solving linear system $Ax = b$ by bringing augmented matrix, to an upper triangular form and then obtain a solution by backward substitution method.

Ex: Solve the linear equation by Gauss elimination method.

$$b + c = 2$$
$$2a + 3c = 15$$
$$a + b + c = 3$$

→ We will write augmented matrix as:

$$\left[\begin{array}{ccc|c} 0 & 1 & 1 & 2 \\ 2 & 0 & 3 & 15 \\ 1 & 1 & 1 & 3 \end{array}\right]$$

→ Interchange 1st & 2nd equation

$$2a + 3c = 15$$
$$b + c = 2 \quad \Rightarrow \quad \left[\begin{array}{cccc} 2 & 0 & 3 & 15 \\ 0 & 1 & 1 & 2 \\ 1 & 1 & 1 & 3 \end{array}\right]$$
$$a + b + c = 3$$

→ Divide first equation by 2.

$$\begin{aligned} a + \tfrac{3}{2}c &= \tfrac{15}{2} \\ b + c &= 2 \\ a + b + c &= 3 \end{aligned} \Rightarrow \begin{bmatrix} 1 & 0 & 3/2 & 15/2 \\ 0 & 1 & 1 & 2 \\ 1 & 1 & 1 & 3 \end{bmatrix}$$

→ Multiply first equation by -1 and add it to third equation.

$$\begin{aligned} -a - \tfrac{3}{2}c &= -\tfrac{15}{2} \\ + \quad a + b + c &= 3 \\ \hline b - \tfrac{1}{2}c &= -\tfrac{9}{2} \end{aligned}$$

→ $$\begin{aligned} a + \tfrac{3}{2}c &= \tfrac{15}{2} \\ b + c &= 2 \\ b - \tfrac{1}{2}c &= -\tfrac{9}{2} \end{aligned} \Rightarrow \begin{bmatrix} 1 & 0 & 3/2 & 15/2 \\ 0 & 1 & 1 & 2 \\ 0 & 1 & -1/2 & -9/2 \end{bmatrix}$$

→ Multiply 2^{nd} equation by -1 and add it to 3^{rd} eqn

$$\begin{aligned} a + \tfrac{3}{2}c &= \tfrac{15}{2} \\ b + c &= 2 \\ b - \tfrac{3}{2}c &= -\tfrac{13}{2} \end{aligned} \Rightarrow \begin{bmatrix} 1 & 0 & 3/2 & 15/2 \\ 0 & 1 & 1 & 2 \\ 0 & 0 & -3/2 & -13/2 \end{bmatrix}$$

→ Multiply 3rd equation by $-2/3$.

$$a + \frac{3}{2}c = 15/2 \Rightarrow \begin{bmatrix} 1 & 0 & 3/2 & 15/2 \\ 0 & 1 & 1 & 2 \\ 0 & 0 & 1 & 13/2 \end{bmatrix}$$

$$b + c = 2$$

$$\boxed{c = 13/3}$$

→ 3rd eqn gives $c = 13/3$. if we Substitute this value in 2nd equation then

$$b + 13/3 = 2$$

$$b = 2 - 13/3$$

$$\boxed{b = \frac{-7}{3}}$$

– Now if we put value of $c = 13/3$ in 1st eqn

$$a + 3/2\, c = 15/2$$

$$a + \frac{3}{2}(13/3) = 15/2$$

$$a + \frac{13}{2} = \frac{15}{2}$$

$$\boxed{a = 1}$$

Scanned by CamScanner