Deterministic Push Down Automata

In the Deterministic Push Down Automata has at most one choice means zero or one no more than one.

A Dpda $M = \langle Q, q_0, \Sigma, \delta, F, \tau, Z \rangle$ is some on npda except the following two condition.

(1) $\delta(q, a, b)$ has ATMOST 1 choice no more than one.

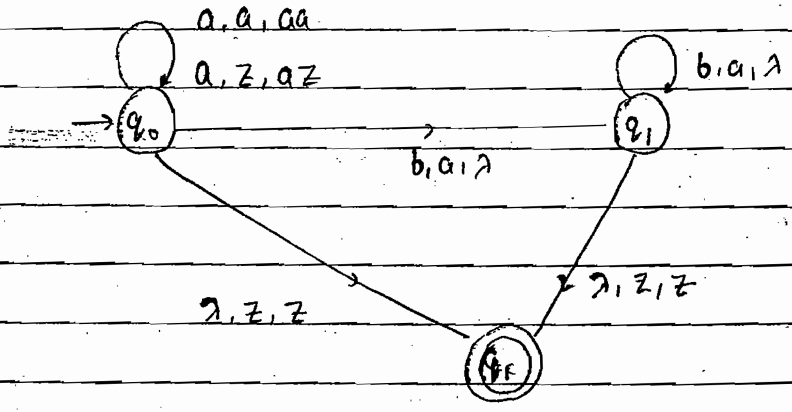
(b) If $\delta(q, \lambda, b)$ is defined then $\delta(q, c, b)$ is empty $\forall\ c \in \Sigma$ (not defined)

→ If in npda λ transition is perform then cannot assign the transition to other symbol.

$$\delta : Q \times \Sigma \cup \{\lambda\} \times \tau \longrightarrow Q \times \tau^*$$

$\delta(q_0, \lambda, b)$	and	$\delta(q_0, c, b)$	Wrong
$\delta(q_0, \lambda, b)$	and	$\delta(q_0, \lambda, c)$	Right
$\delta(q_0, \lambda, b)$	and	$\delta(q_0, c, a)$	Right

e.g. $L = \{a^n b^n \mid n \geq 0\}$

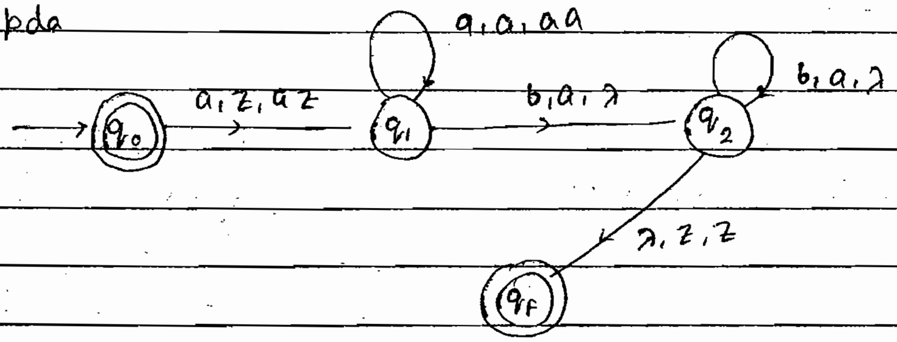


$$\left.\begin{aligned}
\delta(q_0, a, z) &= \{(q_0, aZ)\} \\
\delta(q_0, a, a) &= \{(q_0, aa)\} \\
\delta(q_0, b, a) &= \{(q_1, \lambda)\} \\
\delta(q_1, b, a) &= \{(q_1, \lambda)\} \\
\delta(q_1, \lambda, z) &= \{(q_f, z)\} \\
\delta(q_0, \lambda, z) &= \{(q_f, z)\}
\end{aligned}\right]$$ These transition breaking the rule 2

So this machine is not valid for dpda.

Every dpda is by default ndpda.

Dpda

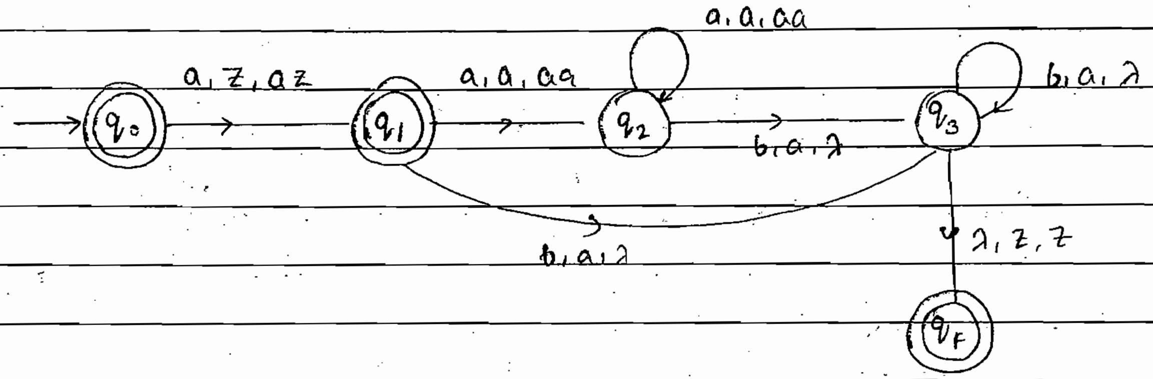


$$\begin{aligned}
\delta(q_0, a, z) &= \{(q_0, aZ)\} \\
\delta(q_1, a, a) &= \{(q_1, aa)\} \\
\delta(q_1, b, a) &= \{(q_2, \lambda)\} \\
\delta(q_2, b, a) &= \{(q_2, \lambda)\} \\
\delta(q_2, \lambda, z) &= \{(q_f, z)\}
\end{aligned}$$

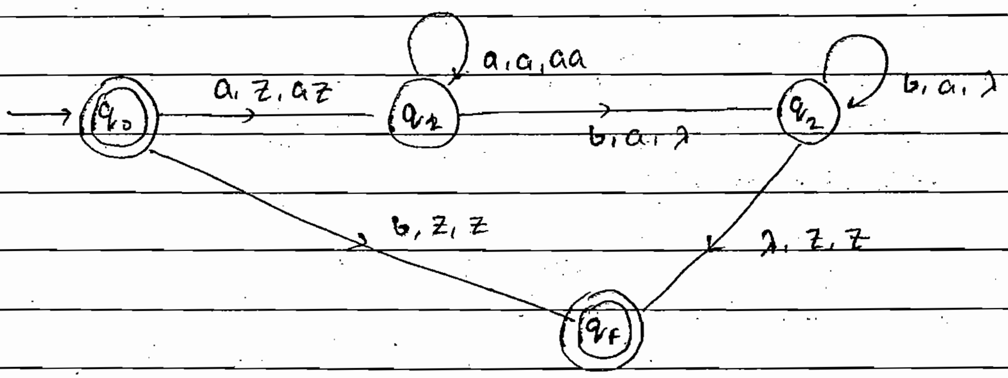
This machine is dpda

This language is $L = \{a^n b^n \mid n \geq 0\}$ is Diterministic C.F.L as well as Non-Diterministic C.F.L

① $L = \{a^n b^n \mid n \geq 0\} \cup \{a\}$

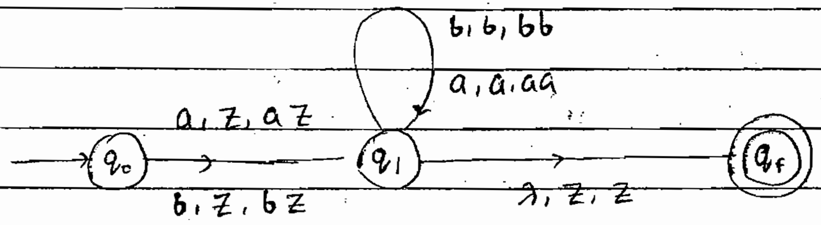


② $L = \{a^n b^n \mid n \geq 0\} \cup \{b\} \quad = \{ b, \lambda, ab, a^2b^2, \ldots$



③ $L = \{ w \in \{a, b\}^* \mid N_a(w) \neq N_b(w) \}$

$L = \{ w \in \{a,b\}^* \mid N_a(w) = N_b(w) \}$



$L = \{ ww^R \mid w \in \{a,b\}^* \}$ not a diterministic C.F.L so we can't make a dpda.

$L = \{ w \in \{0,1\}^* \mid$ w is a palindrome

$\{ w \mid w = w^R$ & $|w|$ is even$\} \cup$

$\{ w \mid w = w^R$ & $|w|$ is odd $\}$.

Not make dpda

⇒ npda is more powerful than dpda.

$\boxed{\text{n dpda} > \text{dpda} \Rightarrow \text{n.c.f.l} \sqsupset \text{d.c.f.l}}$

① $L = \{ww \mid w \in \{a,b\}^*\}$ is not C.F.L so can't make pda with one stack
We will make it with 2-stack pda.

★ Two stack pda is more powerful than one-stack pda.

② $L = \{a^n b^m c^n d^m \mid n, m \geq 0\}$

Converting a C.F.G to npda

Requirement – C.F.G. has to be in G.N.F.

Steps –

1. Push S (start symbol) on the stack.

$$\delta(q_0, \lambda, Z) = (q_1, SZ)$$

2. $\underset{\substack{\downarrow \\ \text{top of} \\ \text{stack}}}{A} \longrightarrow \underset{\substack{\downarrow \\ \text{reading} \\ \text{the symbol}}}{a}\underset{\searrow\ \text{replacement}}{x} \qquad A \in V, \quad a \in T, \quad x \in V^*$

3. $\delta(q_1, a, A) = (q_1, x) \qquad [x \text{ can be } \lambda]$

4. $\delta(q_1, \lambda, Z) = (q_F, \cancel{Z})$

ex- $S \longrightarrow aSb$

$S \longrightarrow ab$

$S \longrightarrow aSB$

$S \longrightarrow aB$

$B \longrightarrow b$

1. $\delta(q_0, \lambda, Z) = \{(q_1, SZ)\}$

2. $\begin{cases} \delta(q_1, a, S) = \{(q_1, SB)\} \\ \delta(q_1, a, S) = \{(q_1, B)\} \end{cases}$

Combine both

$\delta(q_1, a, S) = \{(q_1, SB), (q_1, B)\}$

$\delta(q_1, b, B) = \{(q_1, \lambda)\}$

Asymptotic Natation : Machine ~~Natation~~ Independent

A straight line which touches the perticular curve on the infinite

1. Big 'O'
2. Big 'Ω'
3. Theta (θ)
4. little 'o'
5. little 'ω'

- Abstract representation of time complexity
- It (time complexity) depends upon
- The dominant term in the expression.

e.g. $5n^2 + 6n + 3 = f(n)$ n = input size

Growth rate depend on the dominant term (Hightest term) $5n^2$.

$f(n) / g(n)$ → Time Complexity (Total count) of A
↳ Execution time.

Assumption - ⊙ $f(n)$ & $g(n)$ both are non - negative because time is always +ve

⊙ $f(n)$ & $g(n)$ both are increasing function of n

⊙ n is very large number.

◆ Big 'O' – • Represents the WORST case behaviour of the algo.

- Gives the UPPER BOUND of the running tim of the algo.

Page No.

Date

Upper Bound : Maximum time but the algo can take for any value of n (Input size)

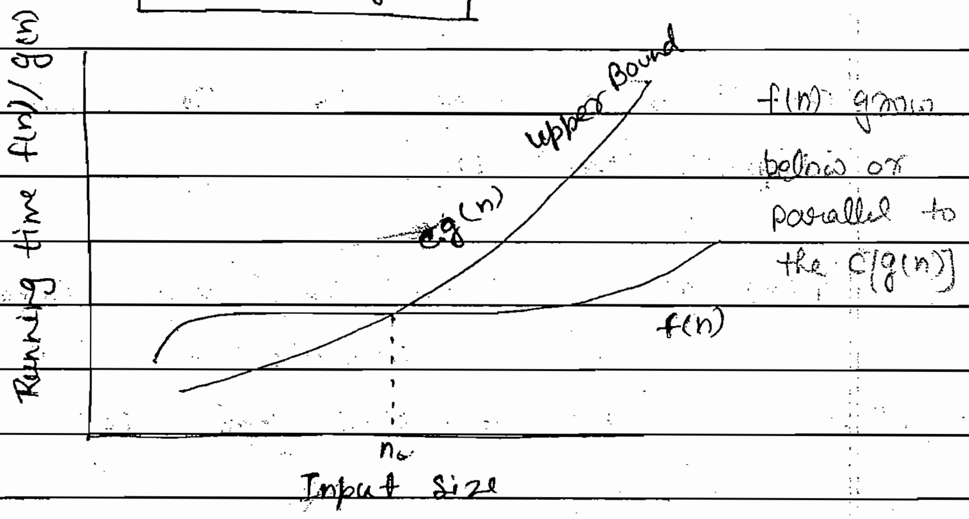
★ $\boxed{f(n) = O(g(n))}$ the value of g(n) is more than OR equal to the value of f(n)

→ $n^2 + 3 = O(n^2)$ TRUE complexity depends on n^2

→ $5n^2 + 3n + 6 = O(n^2)$ TRUE dominating term

There exists two +ve constants c and n_0 such that

$\boxed{f(n) \leq c\,g(n)}$ $\forall\, n \geq n_0$



e.g $f(n) = 5n^2 + 3n + 6$ represent in terms of Big O notation.

The growth rate is depend on the higher degree term.

$5n^2 \mp 5n^2$

$3n \leq 3n^2$

$6 \leq 6n^2$ $n \geq 1$

$5n^2 + 3n + 6 \leq 5n^2 + 3n^2 + 6n^2$ $\boxed{n \geq 1}$

when compar[illegible], ignore [illegible]
then compare

$f(n) \leq 14n^2$ $\quad n \geq 1$

$c = 14$

$g(n) = n^2$

$$\boxed{f(n) = O(g(n)) = O(n^2)}$$

Limit Representation

$$\Rightarrow \boxed{\lim_{n \to \infty} \frac{f(n)}{g(n)} = 0 \;/\; \text{finite quantity (Equal value)}}$$

e.g. $n = O(n^2)$

$$\lim_{n \to \infty} \frac{f(n)}{g(n)} = \lim_{n \to \infty} \frac{n}{n^2} = \lim_{n \to \infty} \frac{1}{n}$$

$$= \frac{1}{\infty} = 0$$

e.g. $2^n = O(2^{n+1})$

$$\lim_{n \to \infty} \frac{f(n)}{g(n)} = \lim_{n \to \infty} \frac{2^n}{2^{n+1}} = \lim_{n \to \infty} \frac{1}{2} = \frac{1}{2}$$

e.g. $2^{n+1} = O(2^n) \Leftrightarrow 2^n \cdot 2^1 = O(2^n)$

$$\lim_{n \to \infty} \frac{2^{n+1}}{2^n} = 2$$

e.g. $(n+a)^b = O(n^b)$

$$\lim_{n \to \infty} \frac{(n+a)^b}{n^b} = \lim_{n \to \infty} \left(\frac{n+a}{n}\right)^b$$

$$= \lim_{n \to \infty} \left(\frac{n}{n} + \frac{a}{n}\right)^b = \lim_{n \to \infty} \left(1 + \frac{a}{n}\right)^b$$

$$= \left(1 + \frac{a}{\infty}\right)^b = (1+0)^b = 1$$

$x^{\infty} = \infty$

$\rightarrow (2^2)^n = 4^n$.

1. $2^{2n} = O(4^n)$ TRUE

$\lim_{n\to\infty} \frac{2^{2n}}{4^n} = \lim_{n\to\infty} 2^{2n} \cdot 4^{-n}$

Ans

UGC Net Dec 201

2. $\log n = O(n)$ TRUE

Ans

3. $\sqrt{n} = O(n)$ TRUE

4. $n! = O(n \log n)$ FALSE

Net June 2014

Big 'O' estimate for factorial function & logarithm of factorial function i.e n! & log n!

$n! = n \times (n-1) \times (n-2) \times \cdots\cdots 3 \times 2 \times 1 \quad n \geq 1$

$n! \leq n \times n \times n \times \cdots\cdots n \times n \times n$

$n! \leq n^n$

$n! < n^n \qquad \boxed{n! = O(n^n)}$

Taking log both sides

$\log n! < \log n^n$

$\log n! < n \log n \qquad \boxed{\log n! = O(n \log n)}$

Ans. $O(n^n)$ & $O(n \log n)$

UGC Net June 2013

Give a good Big 'O' estimate for the following function.

$(n \log n + n^2)(n^3 + 2)$ & $(n! + n^2)(n^3 + \log(n^2 + 1)$

e

$n^2 \cdot n^3 = O(n^5)$ & $n! \cdot n^3 = O(n! \, n^3)$

Ans $O(n^5)$ & $O(n^3 . n!)$

Ugc Net Dec 2013

What is the Big 'O' estimate for

$f(x) = (x+1)\log(x^2+1) + 3x^2$

$x \log x^2 + 3x^2 = 2x \log x + 3x^2$

Ans. $f(x) = O(x^2)$

- Big 'Ω'

- It represents the best case behaviour of al
- It gives the lower Bound on the running time of algo.

$\boxed{f(n) = \Omega\, g(n)}$ g(n) is smaller or equal to f(n)

$n \geq 1$

$n = \Omega(n)$

$n^2 = \Omega(n^2)$

$n^2 = \Omega(n)$

$\log n = \Omega(\log \log n)$ TRUE

$n = \Omega(\sqrt{n})$

(n)

$n^2 + 2n + 5 = \Omega(n^2)$

$2^n = \Omega(4^n)$ False

There exist two +ve constants c & n_0 such that

$\boxed{f(n) \geq c \cdot g(n)}$ $\forall\, n \geq n_0$

$n^2 + 1)$

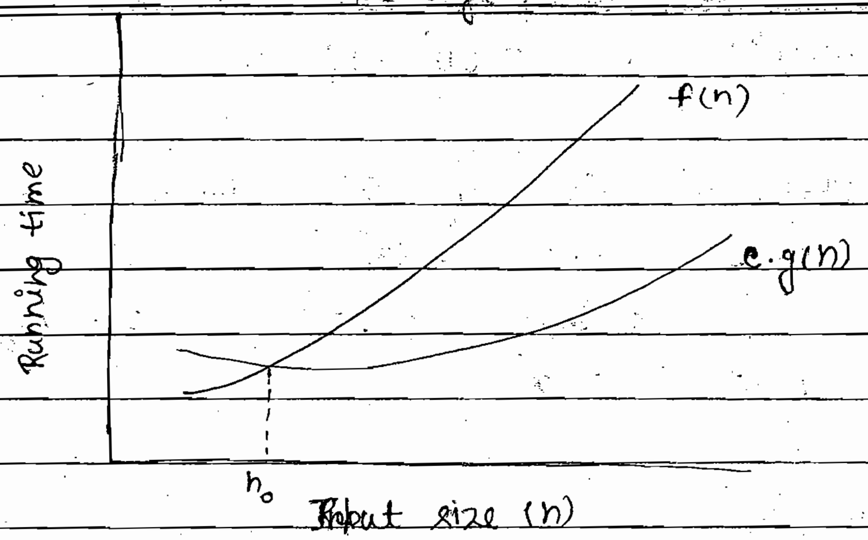
eg $2^{2n} = O(2^{n+1})$ false

$2^{2n} = 2^n \cdot 2^1$

$,3)$

$2^{2n} > 2^n$

growth rate of f(n) is above of the c.g(n).



eg. $f(n) = 5n^2 + 6n + 3$

$= \Omega(n^2)$

$f(n) = 5n^2 + 6n + 3 > 5n^2 \quad n \geq 0$

$f(n) > 5n^2 \quad \forall\ n \geq 0$

$c = 5 \ , \ n_0 = 0$

$g(n) = n^2$

eg. $f(n) = n^4 + 3n^3 + 4n^2 + 10$

$= \Omega(n^4)$

$f(n) = \Omega(g(n))$

$$\boxed{\lim_{n\to\infty} \frac{f(n)}{g(n)} = \infty / a \quad \text{finite Quantity other than } 0}$$

$n! = 1 \times 2 \times 3 \times ---- (n-1) \times n$

$= O(n^n)$

$n! = 1 \times 2 \times 3 \times 4 ...... (n-1) \times n$

$n! > 2 \times 2 \times 2 \times 2 \times ---- 2 \times 2$

$n! > 2 \times 2 \times 2 ---- n$ times

$n! > 2^n$

Sets :→ Set is a "well defined collection of distinct object"

↓

for every person answer must same

(Vowels , States of India)

$\{a\} = \{a, a\} = \{a, a, a\}$

eg. set of english vowels = {a, e, i, o, u}

= {e, a, o, i, u}

Representation of Set -

1. Roster Form : Set enclosed in { }

elements separated by , .

eg. $A = \{a, e, i, o, u\}$

$B = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$

2. Set Builder Form : A = { variable ∈ Domain : Conditi

: such that 'OR' | such that

eg. A = { x ∈ english alphabet : x is vowel }

eg. $A = \{x \in N : -1 \leq x \leq 3\}$

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

eg. $A = \{x : x^2 - 25 = 0\}$

$A = \{-5, 5\}$

Evolution of Number System

- Natural Number (N) / Counting Number —

$N = \{1, 2, 3, \dots\dots\}$

- Whole Number (W) —

$W = \{0, 1, 2, 3, \dots\dots\} = \{0, N\}$

Zero — Zerifo "How can nothing be something"

- Integer Number (I) OR (Z) [Zahlen] German words —

$I = \{\dots\dots -3, -2, -1, 0, 1, 2, 3 \dots\dots\}$

- Set of Rational Number (Q) — [Quotient]

$Q = \{p/q : p, q \in Z, q \neq 0\}$

Repeating or terminating decimal Number.

e.g. $10/3 = 3.33$ Repeating decimal number
$= 3.\bar{3}$

e.g. $5/2 = 2.5$ Terminating decimal number

- Irrational Number Neither repeating nor terminating

e.g. $\pi = 3.14\dots$, $\sqrt{2}$, $\sqrt{3}$

Page No.

Date

$$\boxed{\mathbb{N} \subset \mathbb{W} \subset \mathbb{Z} \subset \mathbb{Q} \subset \mathbb{R} \subset \mathbb{C}}$$

$\pi \approx 22/7$

π ↓ Irrational

$22/7$ ↘ Rational

e = euler No.

ϕ = Golden Ratio

★ Irrational Numbers used in 'Cryptography'.

- Real Number —

Rational No + Irrational No. = Real No.

- Complex Number —

$C = \{a + \underset{\text{iota [imaginary]}}{\underset{\downarrow}{i}}b : a, b \in R\}$

e.g. $Z = \underset{\text{Real part}}{\underset{\downarrow}{2}} + \underset{\text{Imaginary part}}{\underset{\downarrow}{3i}}$

$$\left\{ \begin{aligned} i &= \sqrt{-1} \\ i^2 &= -1 \\ i^3 &= -i \\ i^4 &= 1 \end{aligned} \right\}$$

Irrational Number — π - pi

→ e - eular No. $= 1 + \frac{1}{1!} + \frac{1}{2!} + \frac{1}{3!} + \cdots\cdots = 2\cdot7$

Natural Logrithm. $2 < e < 3$

→ ϕ - Golden Ratio! $\phi = \frac{F_n}{F_{n-1}}$ ★ $\phi = \frac{1+\sqrt{5}}{2}$

$\boxed{\phi = 1.\cancel{1}61}$, $\phi = \lim_{n \to \infty} \frac{F_n}{F_{n-1}}$

Use in 'Fibonacci Series'/ Human Face

Page No.

Date

Q. The golden ratio ϕ & it's congugate ϕ both satisfy the eq^n.

(a) $x^3 - x - 1 = 0$ ✓(b) $x^2 - x - 1 = 0$

(c) $x^3 + x - 1 = 0$ (d) $x^2 + x - 1 = 0$

$x^2 - x - 1 = 0$

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \Rightarrow \frac{1 \pm \sqrt{5}}{2}$$

Types of Set —

1. Empty Set / Null Set / Void Set : A set which contain no element.

$$\boxed{\{\ \} \quad \text{or} \quad \phi}$$

2. Singleton Set : A set containing only one element.

g. $A = \{a\}$, $B = \{\phi\}$, $C = \{1\}$

$A = \{\{1, 2\}\}$

$\{1\} \neq \{1, \{1\}, \{\{1\}\}\}$

3. Finite Set : A set containing finite number of element.

eg. $A = \{a, e, i, o, u\}$

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

4. Infi

e.g.

Con

'con

set

set

Rep

e.g. A

1F

5. Eq

no

e.g. A

B

C

D

F

E

C

4. Infinite Set : A set which is not finite.

e.g. $\mathbb{N} = \{1, 2, 3 \ldots\}$

$Z = \{\ldots\ldots -2, -1, 0, 1, 2, \ldots\}$

Cardinality / Cardinal No. of Finite Set – n is called a 'cardinal number' or cardinality of a finite set A & it's is the no. of element in set A.

It is also called 'order of set'

Represented by

$$\boxed{|A| = n(A) = n \text{ (no. of element)}}$$

e.g. $A = \{a, e, i, o, u\}$

$|A| = n(A) = 5$

5. Equivalent Set :- Two finite set A & B are equivalent if their cardinal no. are same. i.e.

$$|A| = |B| \text{ or}$$
$$n(A) = n(B)$$

e.g. $A = \{a, b\}$ $\qquad E = \{0, 1\}$

$B = \{1, 2, 3, 4\}$ $\qquad F = \{8, 4, 12\}$

$C = \{4, 8, 12\}$ $\qquad H = \{p, q\}$

$D = \{3, 1, 2, 4\}$ $\qquad G = \{1, 5, 7, 11\}$

$$\left.\begin{array}{l} A \neq E \neq H \\ B \neq D \neq G \\ C \neq F \end{array}\right\} \text{, are equivalent sets.}$$

6. **Equal Set :** Two set A & B are said to be equal if every member of A is a member of B & vice-versa.

e.g. $A = \{a, b\}$, $B = \{b, a\}$ $\boxed{A = B}$

Subset :- Let A & B be two sets. If every element of A is an element of B then A is called subset of B.

Denoted By $A \subseteq B$

i.e. $\boxed{A \subseteq B \,;\, \forall a \in A \Rightarrow a \in B}$ a → element

e.g. $A = \{a, b\}$, $B = \{a, b, c\} \Rightarrow A \subseteq B$

$A = \{d, b\}$, $B = \{a, b, c\} \Rightarrow A \nsubseteq B$

- A is a subset of B $A \subseteq B$

OR

B is a superset of A $B \supseteq A$

$\boxed{A \subset B \Rightarrow A \neq B}$

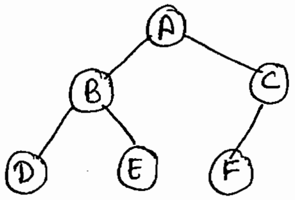
Proper Subset : If $A \subseteq B$ & $A \neq B$ then A is called proper subset of A

i.e. $\boxed{A \subset B}$

16-OCT-2015

BINARY SEARCH TREE :- Binary tree with search property

e.g.



Above tree is BST then

(a) D < B < A < E < C < F

(b) A < B < D < E < C < F

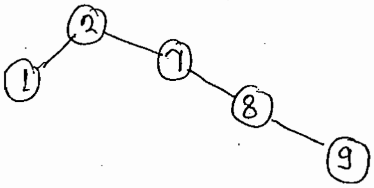
[✓ (c)] D < B < E < A < F < C

(d) Nota

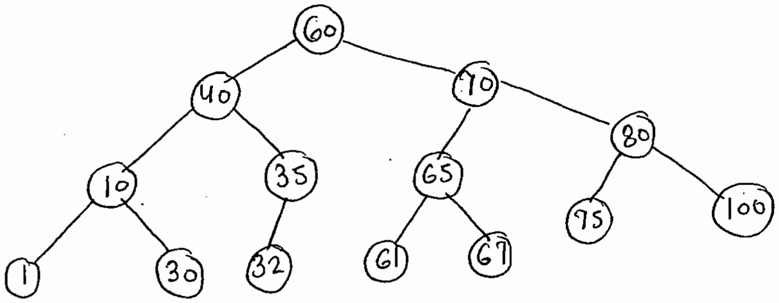
e.g. if 2, 7, 8, 1, 9 is inserted into a BST in that order. How many BST's are possible.

[✓ (a)] 1 (c) 5

(b) 42 (d) Nota



e.g.

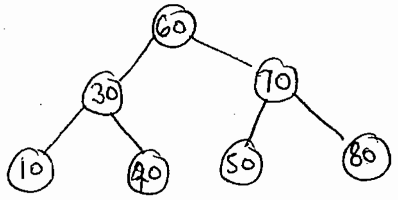


Traverse in Inorder in Binary search tree will always in Increasing Order.

Cost of Search in BST :- O (height)

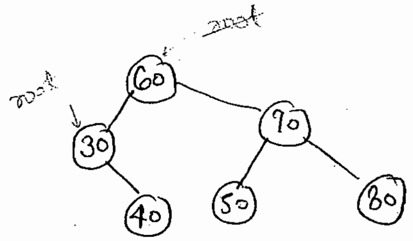
Search 61 cost = O (log n)

Minimum of BST : left most leaf node but not always



Minimum = 10

But if ~



then Minimum will be that node whose left child is NULL in the left subtree.

```
int smallest (tree * root)
{
    if (root != NULL)
    {
        while (root → left != NULL)
            root = root → left;
        return root → data;
    }
}
```

Maximum of BST : Right child in NULL

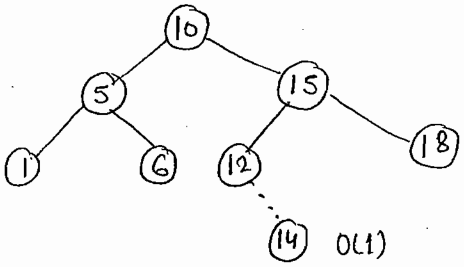
```
int largest (Tree * root)
{
    if (root != NULL)
    {
        while (root → right != NULL)
            root ⇒ root → right;
        return root → data;
    }
}
```

Cost of finding Maximum & Minimum in BROST Case = O(n)

Cost of finding maximum & Minimum is BEST case = O(1)

Inserting a New node in a BST : First find the proper location of searched element.

Insert 14 :

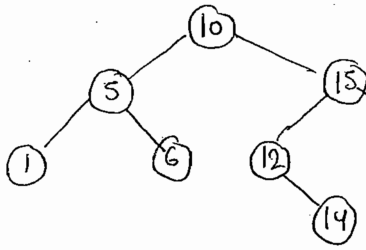


Cost of insertion = Cost of searching + Cost of demand memory

= O (height) + O(1)

Cost (insertion) = O (height of BST)

Deletion in a BST :



Inorder : 1, 5, 6, 10, 12, 14, 15, 20

↑ predecessor ↑ Successor

Inorder predecessor of 12 = 10

Inorder Successor of 12 = 14

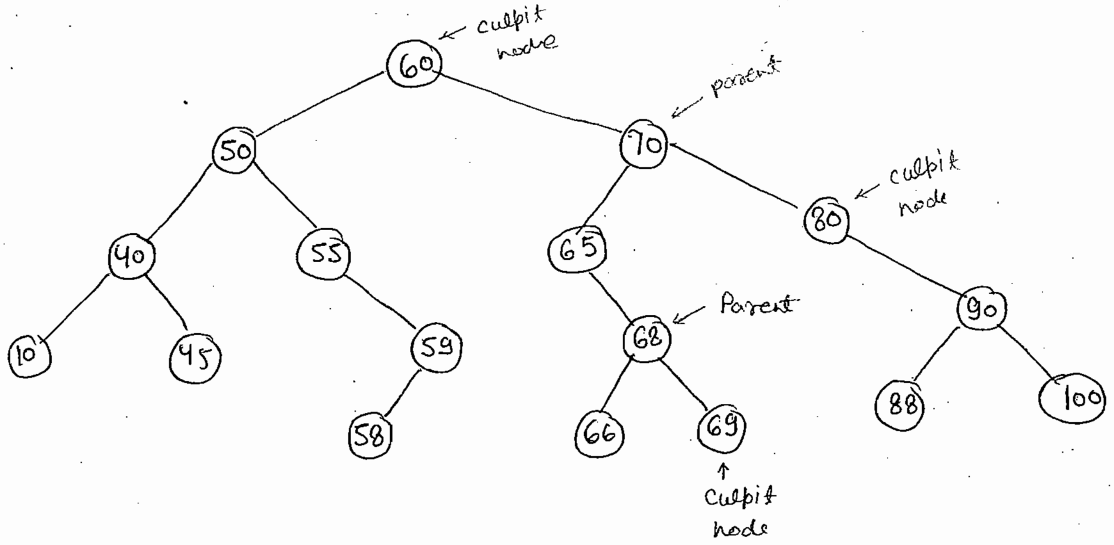
Inorder Predecessor : Is the largest element of the left subtree if it exists.

Inorder Successor : The smallest element of the right sub tree if it exists.

Case - I : When the Culpit (Delete) node is a leaf node.

Case - II . When the culpit node has only one child.

Case - III When the culpit node has both left & right child.



Case I : Delete 69

First find the position of 69

then find the parent of 69

then if it is right child

then remove the right link.

Case II : Delete 80

First traverse the tree & find location

Find parent of culpit node

if deleted node is right child

then link the right child of ~~par~~ culpit node with the right link of parent.

Case - III : Delete 60

It can be deleted by the Predessar or Successor

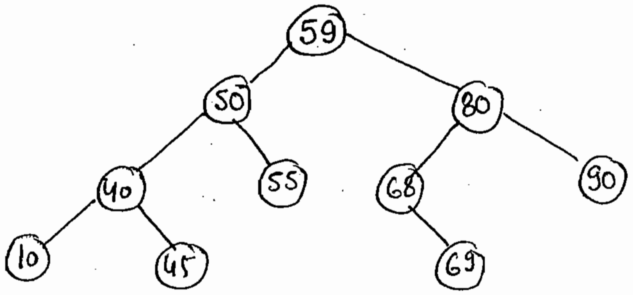
Replace culpit node with it's successor

or then use case I

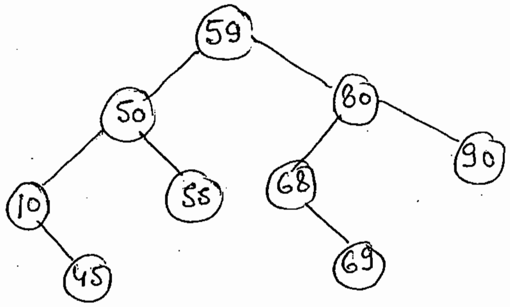
OR

Replace the culpit node with the predecessor & use case II.

e.g.



Delete 40



Delete by predecessor.

Q. If the preorder of a BST is $\overset{\text{root}}{5},\ \overbrace{\underset{\text{root}}{3},\ \underset{l}{1},\ \underset{r}{4}}^{\text{left}},\ \overbrace{\underset{\text{root}}{7},\ \underset{l}{6},\ \underset{r}{8}}^{\text{right}}$ then what is the postorder of BST

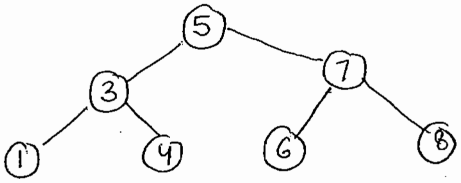
(a) 1, 3, 4, 5, 6, 7, 8

(c) 3, 4, 7, 6, 1, 8, 5

✓(b) 1, 4, 3, 6, 7, 8, 5

(d) Nota

Inorder = 1, 3, 4, $\underset{\text{root}}{5}$, 6, 7, 8



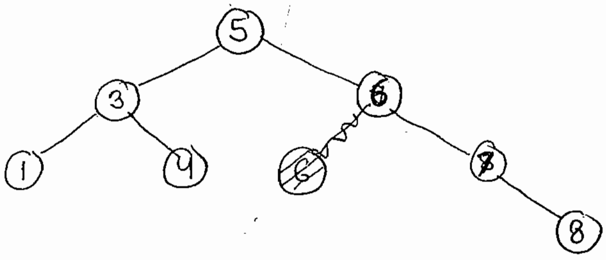
Q. If the BST contains 1, 3, 4, 5, 6, 7, 8 keys then which of the following can be a correct preorder traversal.

(a) 5, 3, 1, 4, 7, 8, 6

(b) 5, 3, 4, 1, 7, 6, 8

✓(c) 5, 3, 1, 4, 6, 7, 8

(d) Nota



Q. If set of data item ranging 10 — 100 is given in a BST & we want to search 55 then which is the current sequence of examing the nodes.

[✓ (a)] 10, 20, 30, 40, 50, 55

(b) 40, 100, 90, 30, 45, 55

[✓ (c)] 80, 70, 60, 45, 55

(d) Nota.

If we search any number then start with any where but check searching element is greater than starting no. so all element is small with.
if starting no. is small then after this all element greater if any, no. is small then sequence is wrong.

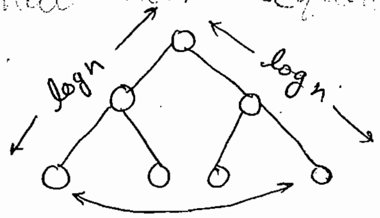
Q. What is the maximum distance B/w 2 nodes in a Perfect Binary tree

(a) logn

(b) 3 logn

[✓ (c)] 2 logn

(d) Nota.



GRAPHICS :- ×Graphics involves storing, displaying and manipulating the picture / image× [Pictorial Represenᵗ

COMPUTER GRAPHICS :- Graphics involves storing, displaying and manipulating the picture / Image and related data for the proper visualization through the computer.

Types of Graphics -

1. Passive Graphics :- User can't change or update the informaᵗ or picture on the console.
e.g. Screen sever

2. Interactive Graphics :- Acc. to the input of user the inforⁿ or picture is changed on console.
e.g. vedio games.
These are widely used

3. Business Graphics :- Use of PPT, Bar chart, π chart

4. Scientific Graphics :- Use of curve fitting, cantoor curve, regression etc.

5. Cartoons & Animation :- Animation is continuous flow of static images.

Applications of Computer Graphics -

1. CAD / CADD :- Use to designe the architecture of Building.

2. Education :– CAL (Computer Aided Learning) , Online learning classes

3. Entertainment :– use graphics in movies

4. Medical :–

5. Science :–

6. Computer Vision :– It is a part of AI . It's aim is design some computer which can understand the real circumstance through images.

7. Image Processing :– Enhance the image (image correction, image reduction is the part of image processing.

Display devices are used to display the picture on the console.

Types of Display devices :–

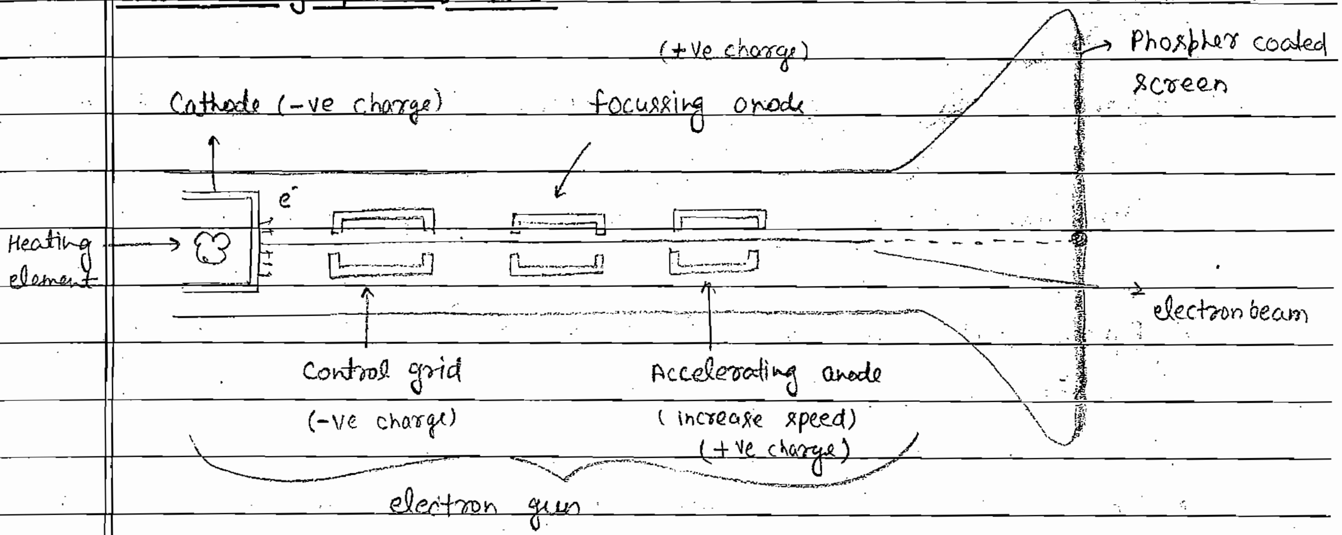
→ DVST (Direct view Storage tube)
→ Calligraphic (Random scan device display divice)
→ Refresh raster display devices

} CRT Commonly used.

→ Flat panel devices → LCD
→ LED

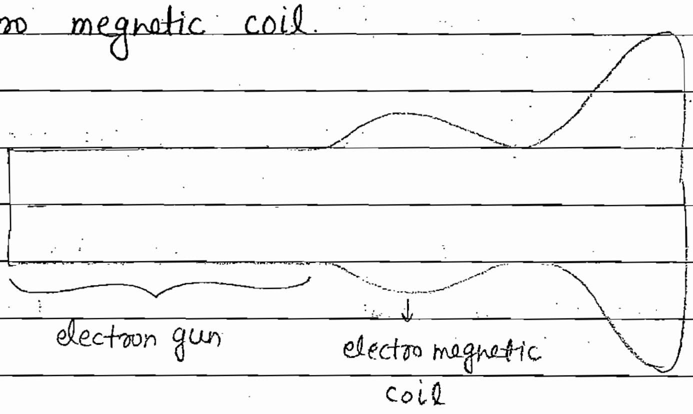
Generation of colors :– → Beam Penetration Method
→ Shadow mask approach or Delta – Delta Approach

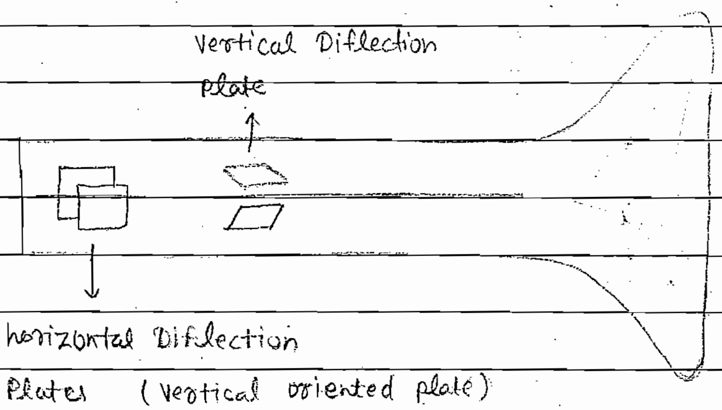
Cathod Ray Tube [CRT] :-



Brightness of screen is controlled by the control grid.
Brightness depends upon the intensity. on electron beam.
Intensity depends upon the no. of electrons.

The electron beam is pointing to a single point on the screen but we want to reflect the electron beam on the whole screen, So we use Diffraction by using electro megnatic coil.




Vertical Diflection plate
horizontal Diflection Plates (Vertical oriented plate)

Display Devices / System

1) Random scan / Calligraphic (straight line) Display System :— It is also known as "Vectored / Stroke / Line Drawing Display system."
(diretion& magnitude)

- Fundamental component of a picture / image is a line (short line). A

- In it we can use any pattern for display.

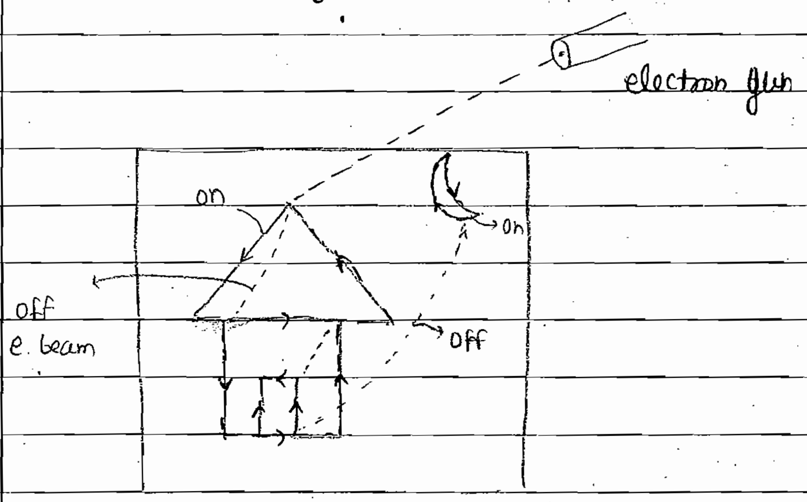
Short persistan a phosphor coated screen & delay + 100ms

In it we have to draw the picture again & again so need to refresh

refresh rate = # of times the picture on the screen is drawn in per unit time

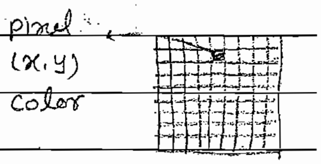
= 30 Hz = 30 fps (frame per sec)

If the refresh rate is less than 30Hz then the picture will be flicking & we can't see the picture clearly.

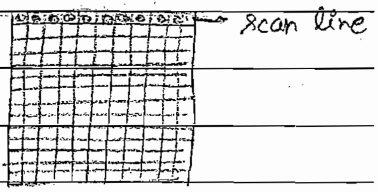


2) Refresh Raster Display system :- This screen is considered as a matrix of pixels

Pixel - 'Picture element' :- Smallest addressable unit / portion on the screen.



pixel contains 3 component -
(x,y) Cordinate
color



M Rows of pixels = m scan line

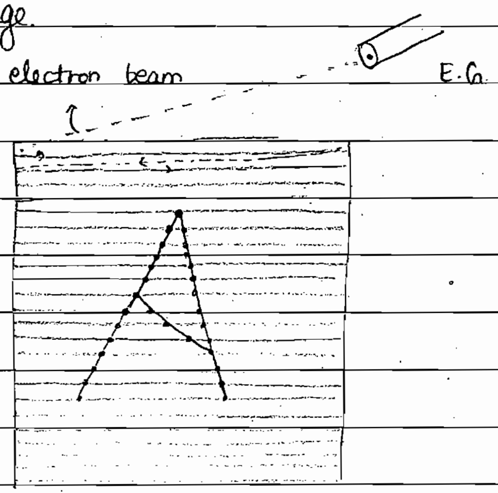
Each Row & column is a scan line.

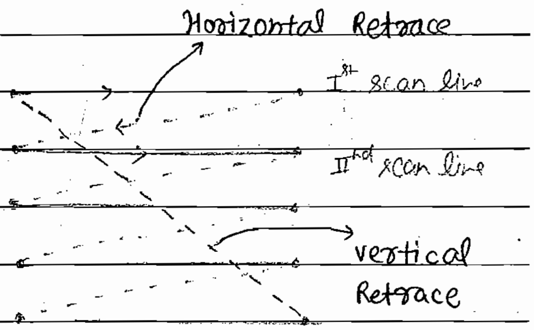
"A picture is a collection of points"

Refresh Buffer - A memory / Buffer the $\underbrace{\text{instruction / program}}_{}$ to display the image on the screen is stored. It is read by the Display Processor & it interprate it & draw the corresponding picture on the screen

Display program

Change in picture, the content of the buffer will be change.





$\boxed{\text{m Scan line} \Rightarrow (m-1) \text{ Horizontal Retrace}}$

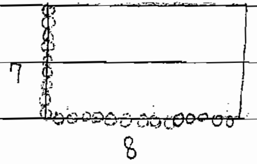
If Referesh rate is = r fps

then $\boxed{\#\text{ of vertical retrace in 1 second} = (r-1)}$

Resolution : # of pixels in unit horizontal direction * # of pixels in unit vertical direction

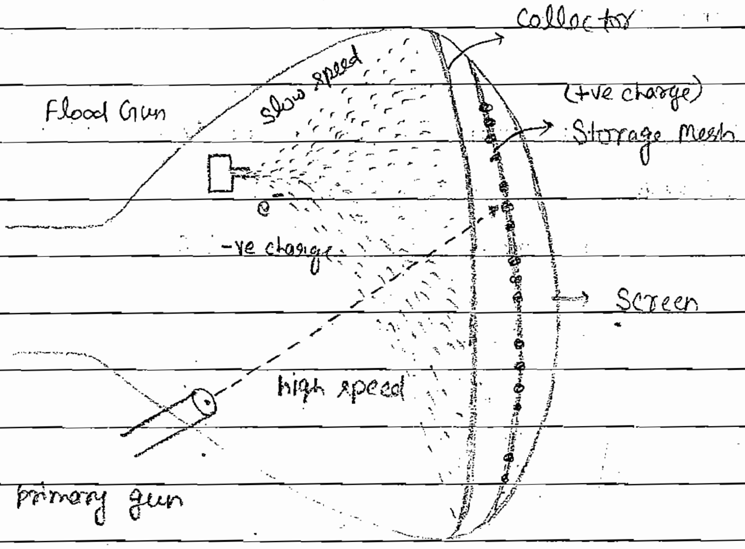
i.e # of pixels in per unit square Area

resolution = 7 x 8
= 56 pixels



3) DVST (Direct view storage tube) − In it long persistence Phosphor coated screen.

No need to refresh so flicker free.



Mobile Computing

5 Ques

CT&T

When computational Devies are movevable in nature.

Communicational Classification : It is three types.

1) 1-G First Generation Mobile – Analog voice is sent . It uses circuit switching .

Bandwidth 0 – 4 KHz

2) 2-G Second Generation Mobile – Digital voice and data, text Packet switching. (virtual connection)

Bandwidth -64 Kbps. [Digital Radio]

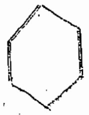
3) 3-G Third Generation Mobile – Digital voice & Data (Text, audio, video.) Packet switching.

→ Telegram uses message switching

Mobile Station (MS) – Moving unit

Base Station (BS) – Stationary unit mobile tower

Cell – The Shape of cell is Hexagonal



large area

Radius of cell 1 to 12 meter

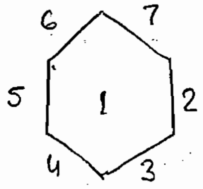
If the Mobile Station moves outside the cell the Roaming will charge MS.

Handoff – If you are moving from one cell another cell.

1) Hard Handoff connection breaked from BS_1 then connection establish to second BS (call drop)

Soft Handoff - Connection of MS establish to second BS then connection will break. from first BS. No call dropping.

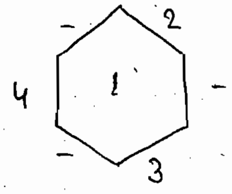
requency Re-Use Factor : (i) Re-Use factor 7



In this case frequency Re-use factor is 7 $\left(\frac{1}{7}\right)$
All cells has diff frequency if any will same then frequency interference can occur.
More frequency Required.

1 cell ← 1 BS

Re-Use Factor 84:



In this case adjecent left blank
frequency Re-use factor is 4 $\left(\frac{1}{4}\right)$

Q - In the following which is frequency re-use factor
(a) N^2 (b) N^3 ✓(c) $1/N$ (d) $\sqrt{N}$

bile Switching Center [MSC] :- Each BS is controlled by switching office called MSC.

rse Communication] or
nsmitting - To place a call from MS, caller enters 7 or 10
s to BS] digits 8 (Phone number) and presses the call button.

. MS scans the band, seeking setup channel with strong gnal & send to closest BS. BS sends that signal to MSC.

r Mobile to Landline communication → MSC sends the (data) signal to telephone office. If the lled party is available then connection is made & result is played back MSC. AT this point MSC assigns an unused voice annel to call & connection is established communication begins.

[forward Comm.]

Receiving - When MS is called, the Telephone central office sends [BS to MS] the number to MSC. MSC searches for the location of MS by sending query signal to each cell in a process called. Paging.

Once Mobile Station is found, the MSC transmits a ringing tone & MS answers, then assigns an unused voice channel to call. Then communication is established.

- First Generation (1G) Mobile : AMPS [Advanced Mobile Phone System] is one of analog cellular system developed by North America. AMPS uses FDMA to seperate links.

AMPS - Transmitting frequency Band - ISM [Intreium Standard]

ISM 800 - MHz

869 - 894 MHz ↙ Receiving bandwidth

824 - 849 Bandwidth = 25

AMPS reuse factor 7

Hard Handoff

Each band~~width~~ is devided into 832 channels

↓

2 provides [each has 416]

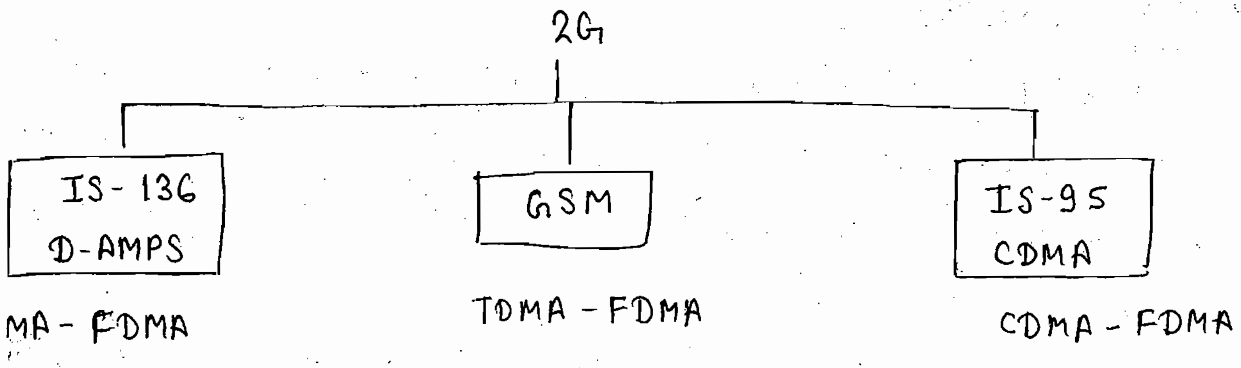
21 channels used for control

395 channels used for calling phase. [416-21] = 395

Disadvantage - Analog voice ~~ratd~~ (bandwidth is low).

& circuit switching is used.

[illegible] .

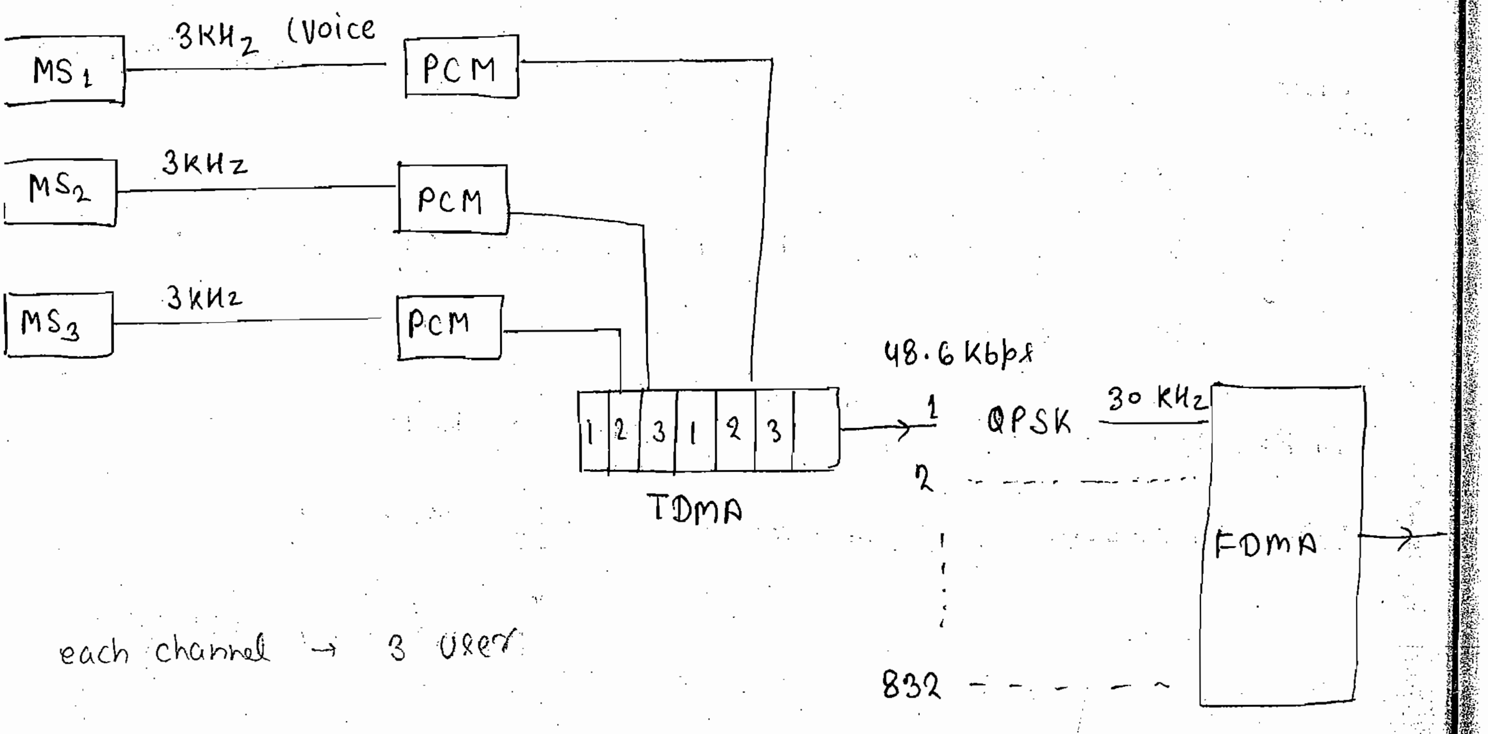


AMPS [Digital AMPS] : Initially D-AMPS denoted by IS - 54 (Interium Standard - 54) and later reveised

- 136 .

quency band → ~~FDM~~ Same as AMPS

h voice channel digitized uses PCM [30KHz → 7.95 Kbps]
[Pluse Code Modulation]



each channel → 3 user

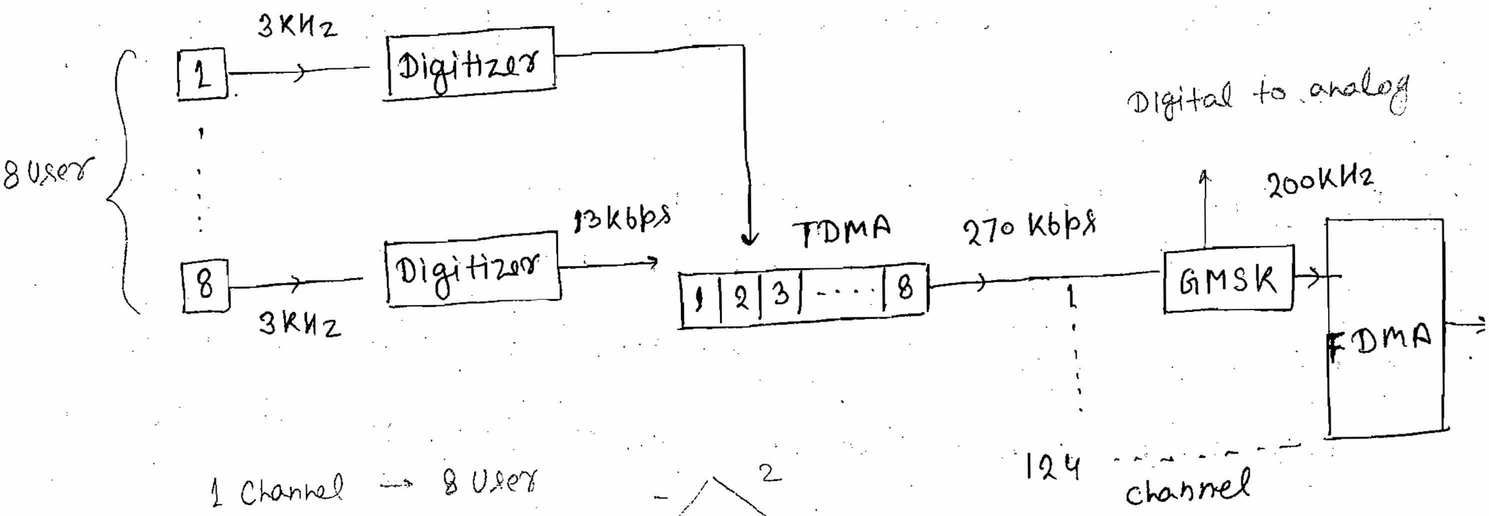
Reuse factor = 7

Standard.

Band - GSM uses two bands for duplex communication. Each band has 25 MHz bandwidth

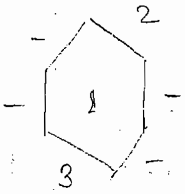
Reverse Band - 124 channel : 890 − 915 MHz

[Receiving] **Forward Band** - 124 Channel : 935 - 960 MHz



1 Channel → 8 User

$\boxed{\text{Re-Use factor} = 3}$

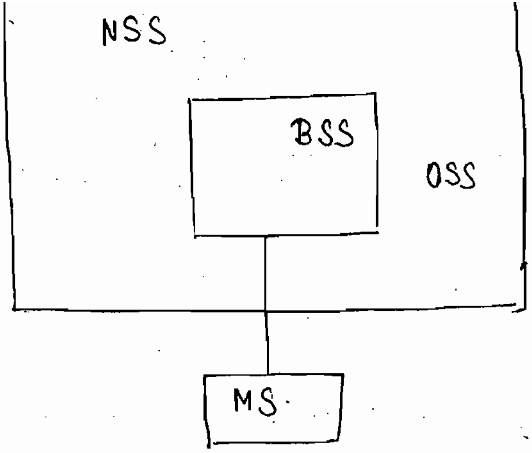


Two left blank.

GSM Architecture :

Terminology used in GSM :

1. MS (Mobile Station)
2. BSS (Base Station Subsystem)
3. NSS (Network switching subsystem)
4. OSS (operation support subsystem)



Optimalty theorm Statement

operation research

The GSM network is devide into the following 3 major component.

SS, BSS, OSS

IS-95 [Interim Standard] :- One of dominant 2G standard in north America.

Band - It works traditional ISM 800 MHz or ISM 1900 MHz

Each band divided into 20 channel of 1.28 MHz sperated by guard band.

Each service provider is allotted 10 channels.

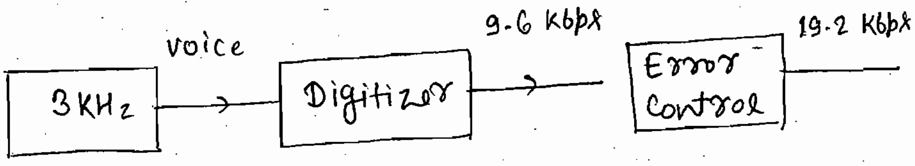
IS-95 is equivalent to 41 AMPS. = 41 X 30 KHz

= 1230 KHz

= 1.23 MHz

Synchronization - By using CDMA synchronization is achieved. To provide synchronization GPS also used

IS-95 forward Transmission : 64 Channel



Operation Research

Linear Programming Problem (LPP)

1) Graphical method → Numerical

2) Simplex method
   (i) two phase method
   (ii) Big-M method

3) Duality

] Theoritical

4) Transportation Problem

5) Assignment problem

} Numerical

Linear Programming Problem (LPP) :

General form of a LPP -

$x_1, x_2, x_3 \; .... \; x_n$ — decision variables

Objective function

$$Z = c_1x_1 + c_2x_2 + \cdots \; c_2x_n \quad \text{———} \; \textcircled{1}$$

$$(c_1, c_2, \cdots c_n) \leftarrow \text{cost vector}$$

Subject to condition,

$$\left.\begin{array}{l} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n \; (\leq, =, \geq) \; b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n \; (\leq, =, \geq) \; b_2 \\ \vdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n \; (\leq, =, \geq) \; b_m \end{array}\right\} \text{—} \textcircled{2}$$

Non-negative constraints,

$$x_1 \geq 0, \; x_2 \geq 0, \; \cdots \; x_n \geq 0 \quad \text{———} \; \textcircled{3}$$

all the decision variable should be Non-negative.

Solution of LPP - Any n-tuple $(x_1, x_2, \cdots x_n)$ which satisfies $\textcircled{2}$ is called a solution to the LPP.

Feasible solution - Any solution of a LPP which also satisfies $\textcircled{3}$ is called a feasible solution.

All values are non negative

Optimal Solution : Any feasible solution which optimizes (max / min) the value of Z is called an optimal solution.

Unbounded Solution : If we can increase / decrease the value of Z infinitely , then we say the problem has unbounded optimal solution

Graphical Solution : when two variable problem

$\underline{\max} \rightarrow Z = 3x_1 + 5x_2$

subject to conditions

$x_1 + 2x_2 = 2000$ $\quad$ $x_1 + 2x_2 \leqslant 2000$

$x_1 + x_2 = 1500$ $\quad$ $x_1 + x_2 \leqslant 1500$

$x_2 = 600$ $\quad$ $x_2 \leqslant 600$

& $\quad x_1, x_2 \geqslant 0$

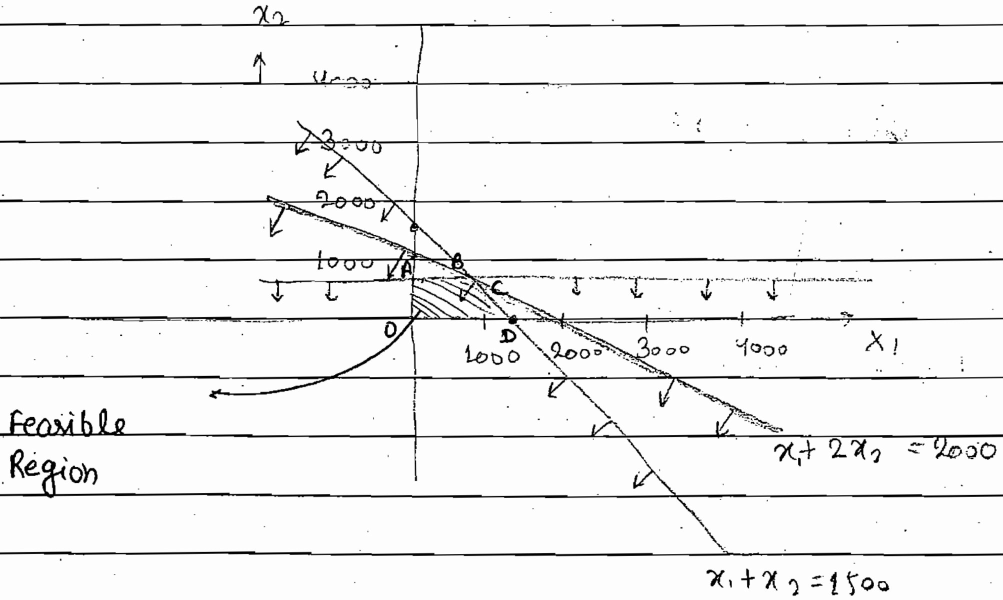
$x_1 + 2x_2 = 2000$

Let $x_1 = 0$

$x_2 = 1000$ $\quad$ $(0, 1000)$

Let $x_2 = 0$

$x_1 = 2000$ $\quad$ $(2000, 0)$



Date :

Page No.

$x_1 + x_2 =$ 1500~~0~~

$x_2 =$ 1500~~0~~	$(0, 1500)$	Let $x_1 = 0$
$x_1 = 1500$	$(1500, 0)$	Let $x_2 = 0$

O	$(0, 0)$	$Z_O = 0$
A	$(0, 600)$	$Z_A = 3000$
B	$(800, 600)$	$Z_B = 5400$
C	$(1000,$ ~~1~~$500)$	$Z_C = 5500$
D	$(1500, 0)$	$Z_D = 4500$

$x_1 + 2x_2 = 2000$

$x_1 + 2 \times 600 = 2000 \quad x_1 \Rightarrow 800$

$$\begin{aligned} x_1 + 2x_2 &= 2000 \\ x_1 + x_2 &= 1500 \\ \hline x_2 &= 500 \\ x_1 &= 1000 \end{aligned}$$

Optimal solⁿ

C $(1000, 1500)$

$x_1 = 1000$ & $x_2 = 1500$

Optimal value

$Z = 5500$

eg Max $Z = x_1 + x_2$

Subject to condition

$x_1 + 2x_2 \leqslant 2000$

$x_1 + x_2 \leqslant 1500$

$x_2 \leqslant 600$

& $x_1, x_2 \geqslant 0$

O	$(0, 0)$	$Z_O = 0$
A	$(0, 600)$	$Z_A = 600$
B	$(800, 600)$	$Z_B = 1400$
C	$(1000,$ ~~1~~$500)$	$Z_C = 1500$
D	$(1500, 0)$	$Z_D = 1500$

C & D gives optimal solⁿ

Above CD all are optimal sol^n

Optimal Sol^n → infinitly many

Optimal value Z → 1500

- If you will get infinitly many optimal sol^n, then objective fu^n line must be parallel to be one of the constraint

e.g. $\text{Min } Z = 1.5x_1 + 2.5x_2$

sub. to constr. $x_1 + 3x_2 \geq 3$

$x_1 + x_2 \geq 2$

$x_1, x_2 \geq 0$

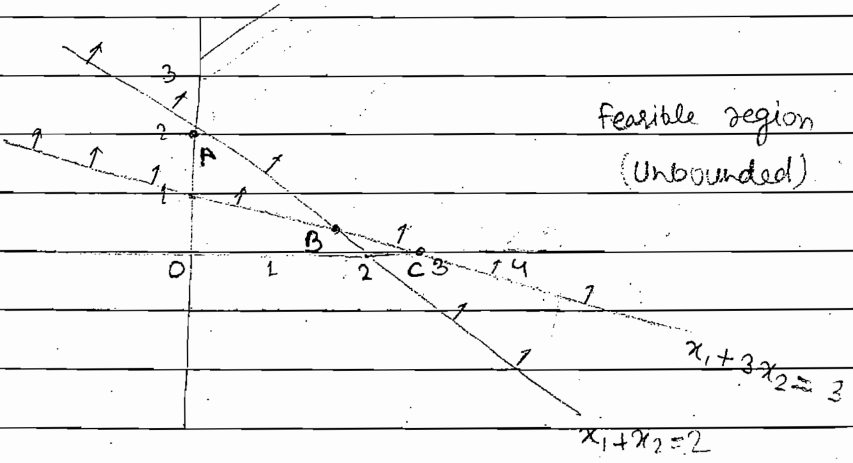
$x_1 + 3x_2 = 3$

Let $x_1 = 0$ Let $x_2 = 0$ ~~(0,3)~~ (0,1)

$x_2 = 1$ $x_1 = 3$ (3,0)

$x_1 + x_2 = 2$ $x_2 = 2$ (0,2)

$x_1 = 2$ (2,0)



Feasible region (Unbounded)

A (0, 2)	$Z_A = 5$
B (1.5, 0.5)	$Z_B = 3.5$
C (3, 0)	$Z_C = 4.5$

$x_1 + 3x_2 = 3$
$x_1 + x_2 = 2$

$2x_2 = 5$
$x_2 =$

Optimal sol^n

$x_1 = 1.5$, $x_2 = 0.5$

Optimal val

$Z = 3.5$ finite value

eg max $Z = 1.5x_1 + 2.5x_2$

$x_1 + 3x_2 \geq 3$

$x_1 + x_2 \geq 2$

$x_1, x_2 \geq 0$

Unbounded optimal sol^n. because region is unbounded.

eg min $Z = -1.5x_1 - 2.5x_2$

$x_1 + 3x_2 \geq 3$

$x_1 + x_2 \geq 2$

$x_1, x_2 \geq 0$

as increase the value of x_1 & x_2, the value of Z will decrease

so it is also unbounded optimal sol^n.

Q. max $Z = 3x_1 + 2x_2$

sub. to

$x_1 - x_2 \leq 1$

$x_1 + x_2 \geq 3$

$x_1, x_2 \geq 0$

$x_1 - x_2 = \cancel{0}\ 1$

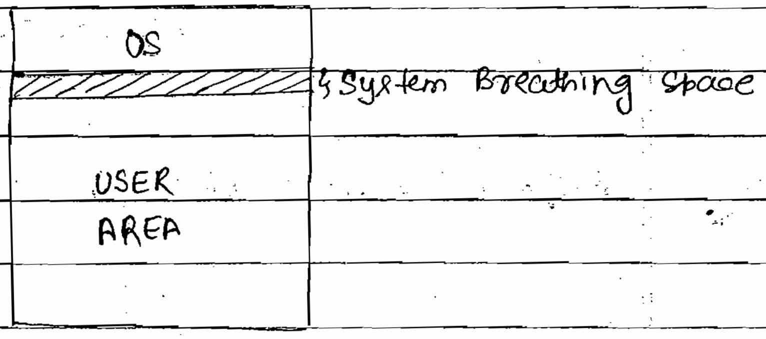
$x_1 = 1$ $x_2 = 0$ (1, 0)

Process State

Non-Preemtive OR Non Preemption

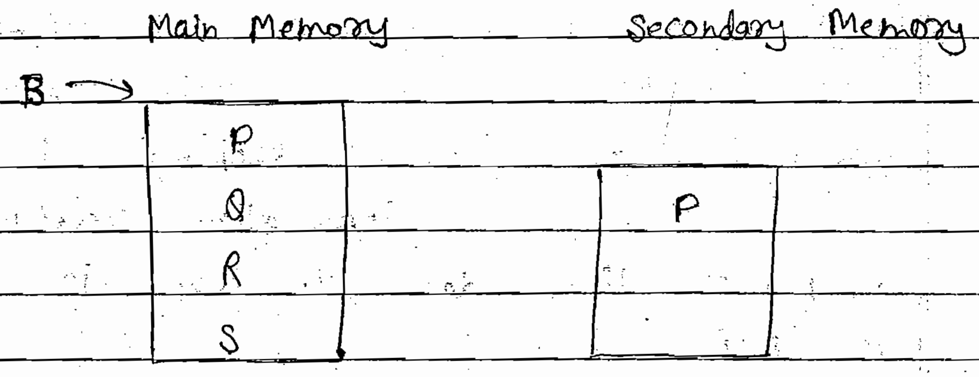
When the process releases CPU voluntarily It is known as Non-Preemption Otherwise Preemption.

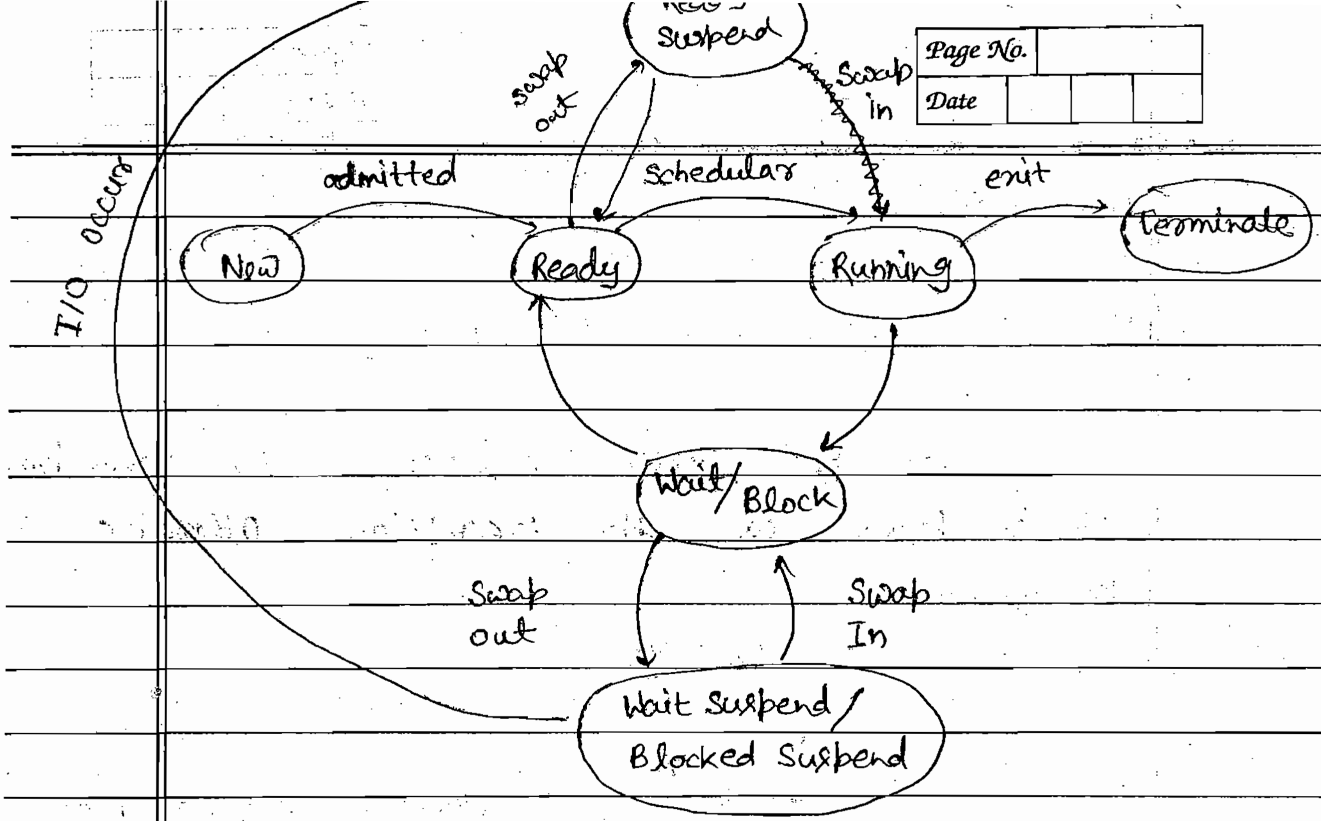
Main Memory



Kernal - That part of OS which resides permanently in main memory is known as Kernal.

Kernal performs necassary operations like Process managment, Process scheduling memory managment & I/O managment.





Suspend State will be in secondary memory.

Q. Consider a system with n CPU then what can be the maximum and minimum no. of processes than may present in ready, running & blocked state

	Min	Max
ready	0	no limit
running	0	n
block	0	no limit

Process Control Block (PCB) — OS maintains a
OR data structure for
Task Control Block (TCB) — keeping all the information related with a process that data structure is known as PCB.

PCB contains the following information —

Process State
Process Id
Program Counter
Register
Memory limit
list of open file
. . .
. . .

- Process State
- Program Counter (A Register tells about the next execute instruction).
- Process id (A unique no. assigned to process
- CPU Register
- CPU Scheduling information
- Memory managment information.
- Accounting information of process
- I/O Status information

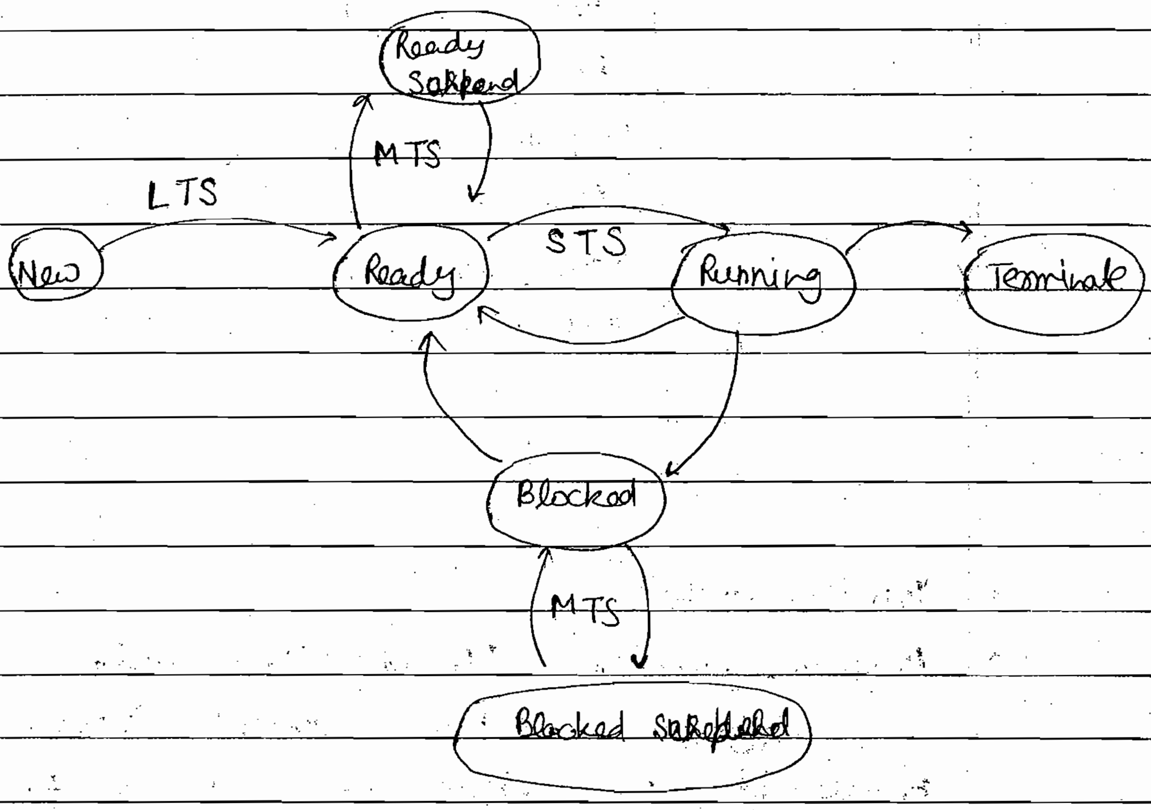
Scheduling Of Process

Schedular — It is a program that schedules the process from source to destination

Types of Schedular —

1. Long term Schedular (very low frequency)
2. Medium term Schedular

3. Short term schedular (more frequency) ⇒



- Long Term Scheduler – The decision to add to the pool of processes to be executed.

- Medium Term Schedular – The decision to add the no. of process that are partially OR full in main memory. 1.

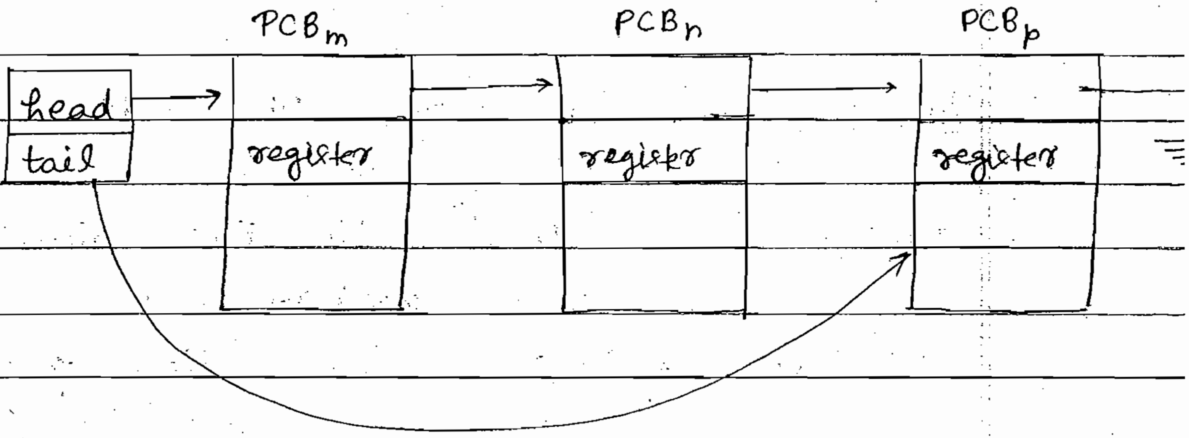
- Short Term Schedular – The decision as to which avialable processes will be executed by the processor. 2.

3.

⇒ STS is also known as 'CPU Schedular'

⇒ The frequency of STS is very high.

Scheduling Queues —



- The process that are residing in main memory & are ready ~~to~~ & waiting to execute are kept in ready queue.

- This ready queue is generally stored as link list as shown in above.

Metrixes Related with Process —

1. Arrival Time (AT) — Arrival time is the time when the process arrive into ready state.

2. Brust Time — Brust time is time required for the process for it's execution.

3. Completion Time — It is time when process completes it's execution

4. Turn Around Time – It is the time difference b/w completion time & Arrival time

TAT = CT – AT

5. Waiting Time = Turn around time – Burst time.

6. Response Time – It is the time difference b/w submission of a job & first response from the system.

CPU Scheduling Policies – / Uniprocessor Scheduling

(1) First Come First Served (FCFS) –: ~~Closest~~

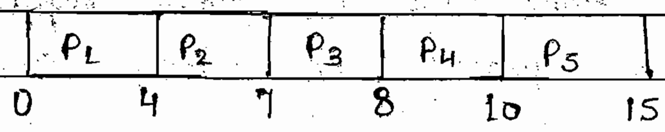
Criteria – Arrival Time

Mode – Non - Preemptive

e.g.

Process No.	AT	BT
1	0	4
2	1	3
3	2	1
4	3	2
5	4	5

Find
(i) Avg WT
(ii) Avg TAT



Gant Chart

1. **Feasibility** — Any Sol^n will follow some constraints & Bounds.

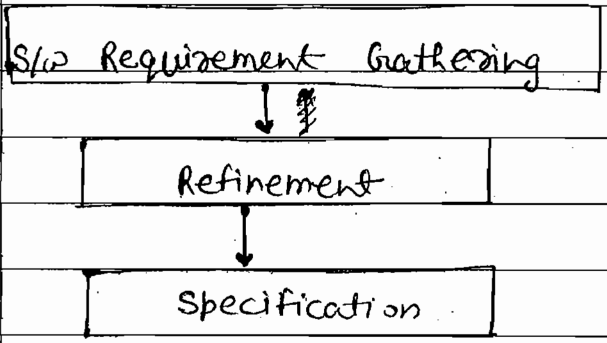
→ Technical Feasibility : Man Power

→ Economical Feasibility : Negotiation b/w client & Service provider.

→ Environmental feasibility :

2. **S/w Requirements** — R

→



→ Functional Requirement

→ Non-functional Requirement

→ Goals of Implementations

3 **S/w Design** — DS + Algo

4. **S/w Coading**/(Implementation)/Development.

5 **S/w Testing** — S/w Debugger is a part of compiler calls system

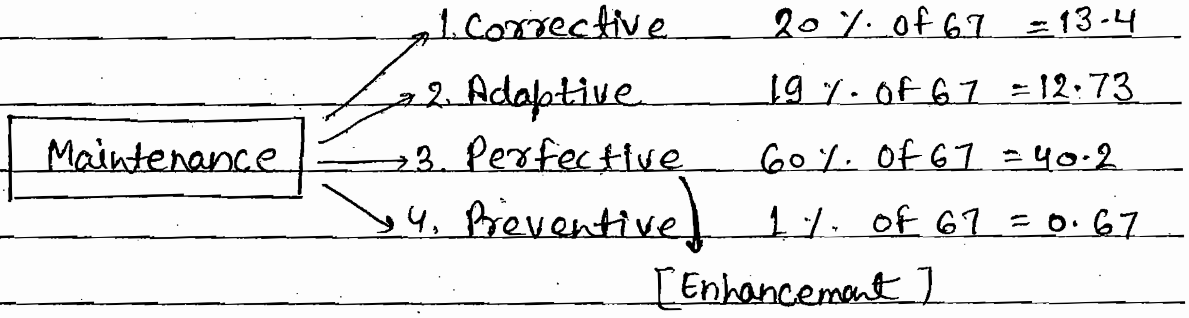
Cycles	Avg Cost (Duration)
S/w Analysis	7%
Design	6%
Coding	5%
Testing	15%
Maintenance	67%

Q. In the following which phase required maximum duration.

(a) Analysis (✓) (b) Design

(c) coading (d) None of these

- S/w always goes under Maintenance.



Maintenance

1. Corrective 20% of 67 = 13.4
2. Adaptive 19% of 67 = 12.73
3. Perfective 60% of 67 = 40.2
4. Preventive 1% of 67 = 0.67

[Enhancement]

Q. In the following which phase required maximum duration.

(a) Analysis (c) Design

(b) Testing (✓) (d) Adaptive Maintenance

Q. Which phase required minimum duration.

(a) Analysis (c) coading

(b) Preventive (✓) (d) None of these.

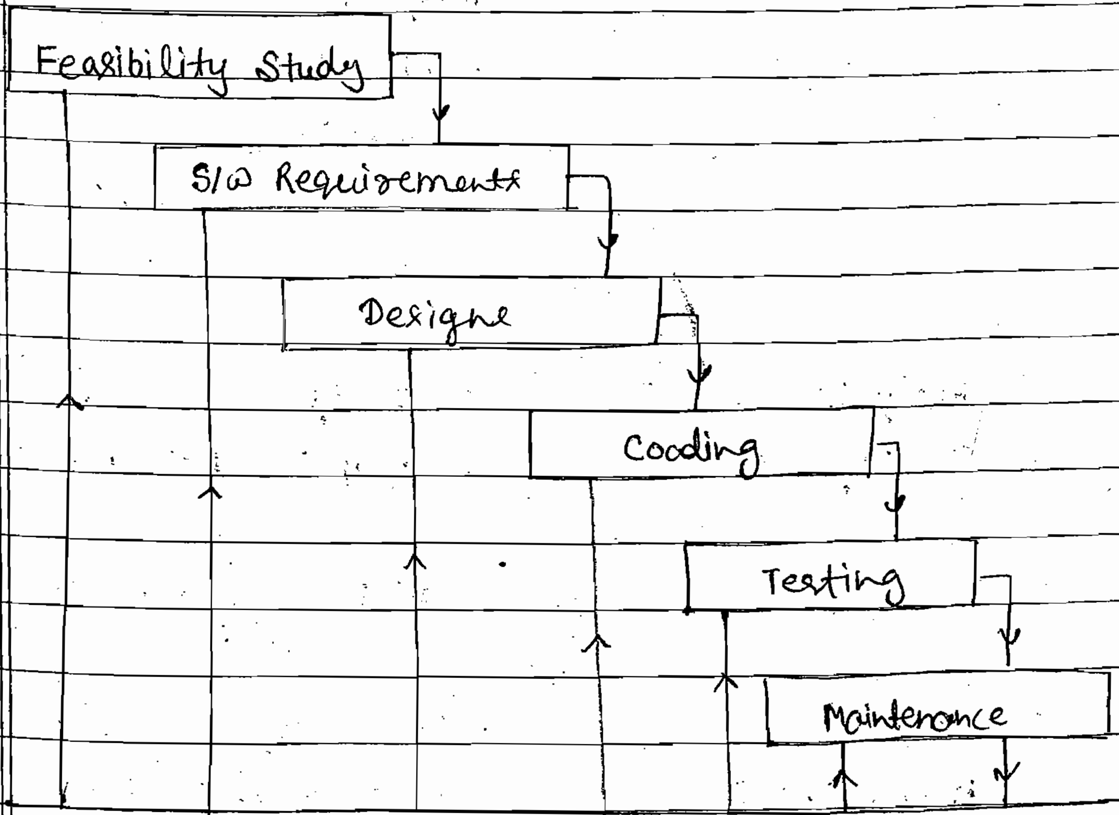
Advantages of Waterfall Model -

1. Simple & Sequential
2. First Model

Short comming / Pitfalls of Waterfall Model -

1. Intially all requirements should be known.
(Requirements are frozen)
2. Heavy document Driven.
3. Error Removal's very costly.
4. Customer Unsatisfaction.
5. Due to many errors S/W Fails.
6. Big - Bang Model [early failure]

(2) Iterative Waterfall Model:



- Verification is done in each phase of SDLC
  → "We are devoloping product right"

- Velidation "we are developing right product"
  ↑ done in last (before delivery of S/w).

- Corrective Maintinance – There are some errors in s/w so corrective maintenance use
  ↗ Remove (fix) residual errors.

- Adaptive Maintinance → For platform independant (Intermediate Code) we use adaptive maintinance

- Perfactive Maintinance [Enhancement] – Initially all requirement are not known. So adding new features we use perfactive.

- Preventive Maintenance - For correcting any error you correct error either in all places or none. [Prevent the change]

When the probablity of error is high than you perform 'Preventive Maintenance'

[Residual Error Removal → Patch : Service Pack]
Remaining

for removing thousands of errors.

Engineering is used"

SOFTWARE -

- Basic unit of ~~System~~ Software is Microinstruction

Microinstruction Interacts with MicroProcessor

- Instruction - One or more than on micro-instruction make Instruction

Used in Architecture

a = b + c

a, b, c microinstruction

+, = instruction

- Program - Set of Instruction.

- Firmware has a unit instruction

- Instruction will represent basic computation of microprocessor

- Instruction represented by binary

- Mircoprocessor does following ~~for~~ on instruction

→ Instruction fetch
→ Instruction Decode
→ Instruction Execute

Engineering is used"

SOFTWARE -

- Basic unit of Software System is Microinstruction

Microinstruction Interacts with MicroProcessor

- Instruction - One or more than on micro-instruction make Instruction

Used in Architecture

a = b + c

a , b , c microinstruction

+ , = instruction

- Program - Set of Instruction.

- Firmware has a unit instruction.

- Instruction will represent basic computation of microprocessor

- Instruction represented by binary

- Mircoprocessor does follwing for on instruction

→ Instruction fetch
→ Instruction Decode
→ Instruction Execute

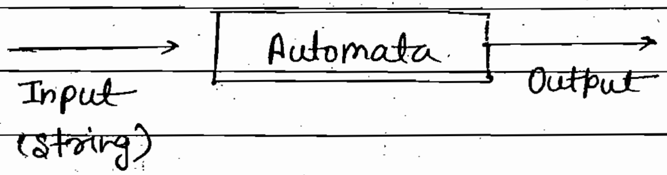
TOC

Motivation & Application :— Construction of Compiler, Text Editor (Regular exp), Pattern Recognition (NFA, Regular Expression).

TOC → Computational Model —

- F.A [Finite Automata)
- P.D.A [Pushdown Automata]
- L.B.A [Linear Bounded Automata]
- T.M. [Turing Machine]

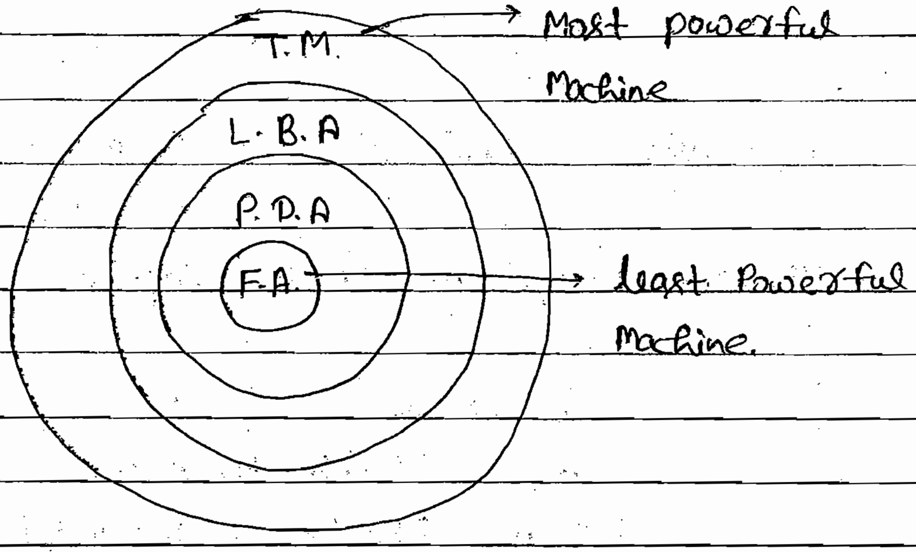
Turing Church Thesis :- " Every work done by digital computer can also be done by T.M."



'Automata' is a fundamental computation machine Which take an input & process without interference gives an output.

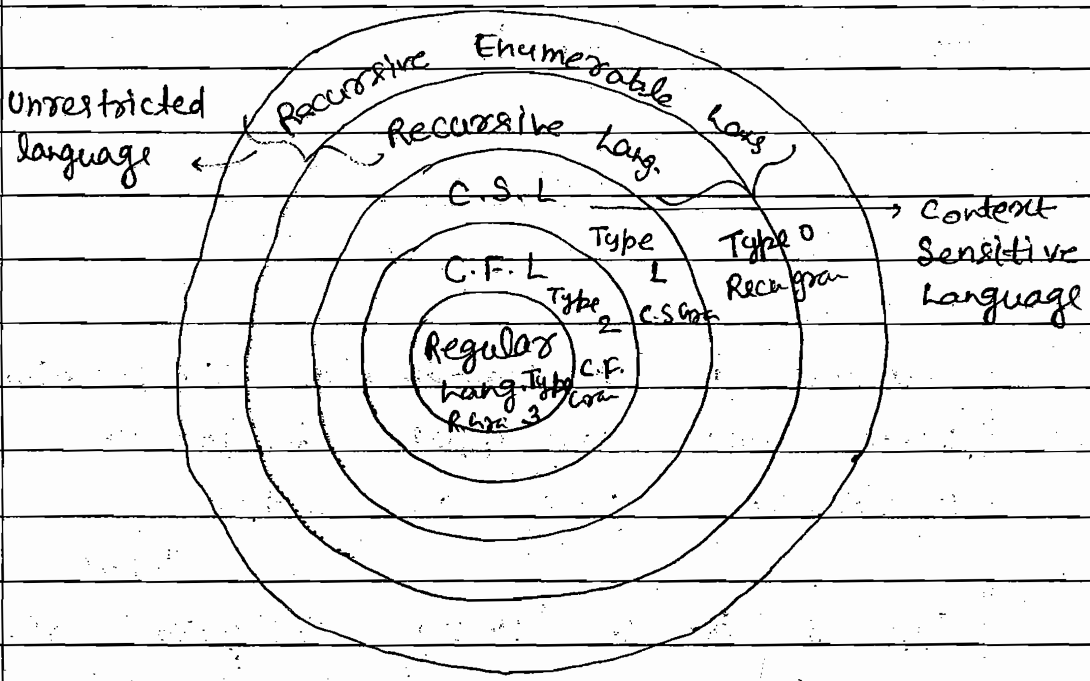
Input will typicall be a 'string'. Collection of string is called 'Language' or 'Formal Language.'

Language (string) may be meaningful or may not be.



Hierarchy w.r. to Power of Machine

$$\boxed{FA < P.D.A < L.B.A < T.M.}$$



Neo chomsky Hierarchy

- All Regular Languges are C.F.L. (Context Free Language).

- Any language outside the boundary will not be Recursive enumerable language.

- Regular languages are accepted by F.A.

- C.F.L are accepted by P.D.A.

- C.S.L are accepted by L.B.A

- T.M use both Recursive lang. & Recursive Enumerable lang.

- Finite Automata is limited within Regular language

Terminal symbols — are denoted by small letters/ digits / special symbols.

like a, b, c, x, y ..... 0, 1, 2, ... #, @, $, ...

Non-Terminal symbols — denoted by capital letters.

Alphabet (Σ) — A finite collection of terminal symbols

e.g. $\Sigma_1 = \{a, b\}$ & $\Sigma_2 = \{0, 1\}$ Binary Alphabate

Word / String — A sequence of symbols taken from Σ.

e.g. $\Sigma = \{a, b\}$

$w = ab$ .. $w = aaaa$

Page No.	
Date	

Operations on the strings / words

1. Finding the length — $\Sigma = \{a, b\}$

$w = abba$

Length of string $\boxed{|w| = 4}$

● Null String / Empty String (λ / ϵ) — A string whose length is '0'

$$\boxed{|\lambda| = 0}$$

2. Concatenation — $\Sigma = \{a, b\}$

$w_1 = abb$, $w_2 = ab$

$$\boxed{\begin{array}{l} w_1 w_2 = abbab \\ w_2 w_1 = ababb \end{array}}$$

$$\boxed{w_1 w_2 \neq w_2 w_1}$$

When $w_1 = w_2$ then

$$\boxed{w_1 w_2 = w_2 w_1}$$

$$\boxed{|w_1 w_2| = |w_2 w_1| = |w_1| + |w_2|}$$

$$\boxed{\lambda w = w \lambda = w}$$

◆ $\boxed{w^0 = \lambda}$

$w^1 = w$

$w^2 = w \cdot w$

$w^3 = w \cdot w^2 = w \cdot w \cdot w$

$\vdots$

$w^n = w \cdot w^{n-1} = w \cdot w \cdot w \cdots$ (n times)

- $\boxed{|w^n| = n * |w|}$

$$[w.w.w.\ldots w] = |w| + |w| + \cdots |w|$$
$$|w^n| = n * |w|$$

3. <u>Reversal</u> – $w = abbbab$

shoe

w^R = Reverse of w

$w^R = babbbab$

$\boxed{\lambda^R = \lambda}$, $\boxed{a^R = a}$ Palindrome

$\boxed{|w| = |w^R|}$ $\boxed{(xy)^R = y^R x^R}$

- <u>Palindrome</u> – $w = aba$

$w^R = aba$

$\boxed{w = w^R}$

→ Even Palindrome – (i) $w = w^R$

(ii) $|w|$ is Even

→ Odd Palindrome – (i) $w = w^R$

(ii) $|w|$ is odd

λ is an Even Palindrome

a is an Odd Palindrome

$\boxed{\text{K}^{th} \text{ symbol from biginiug} = \text{K}^{th} \text{ symbol from last}}$

e.g. # of Palindromes of length 8 over $\Sigma = \{0, 1\}$

$\underline{2}\ \underline{2}\ \underline{2}\ \underline{2}\ \}\ \_\ \_\ \_\ \_$

$2^4 = 16$

e.g. # of Palindromes of length 9 over $\Sigma = \{0, 1\}$

$\underline{2}\ \underline{2}\ \underline{2}\ \underline{2}\ |\ \underline{2}\ |\ \_\ \_\ \_\ \_$

$2^5 = 32$

e.g. # of Palindromes of length n over Σ with $|\Sigma| = K$

$$\boxed{K^{\lceil n/2 \rceil}}$$

Kleen Stare / Kleen Clouser = If $\Sigma = \{a, b\}$

Σ^* = The set of all string which can be constructed by using the symbols from Σ including λ.

e.g. $\Sigma = \{a, b\}$

$\Sigma^* = \{ \underbrace{\lambda}_{0}, \underbrace{a, b}_{1}, \underbrace{aa, bb, ab, ba}_{2}, aaa,$

$bbb, aab, aba, baa, bba, bab, abb,$

$------\infty\}$

↳ Universal Language