AI

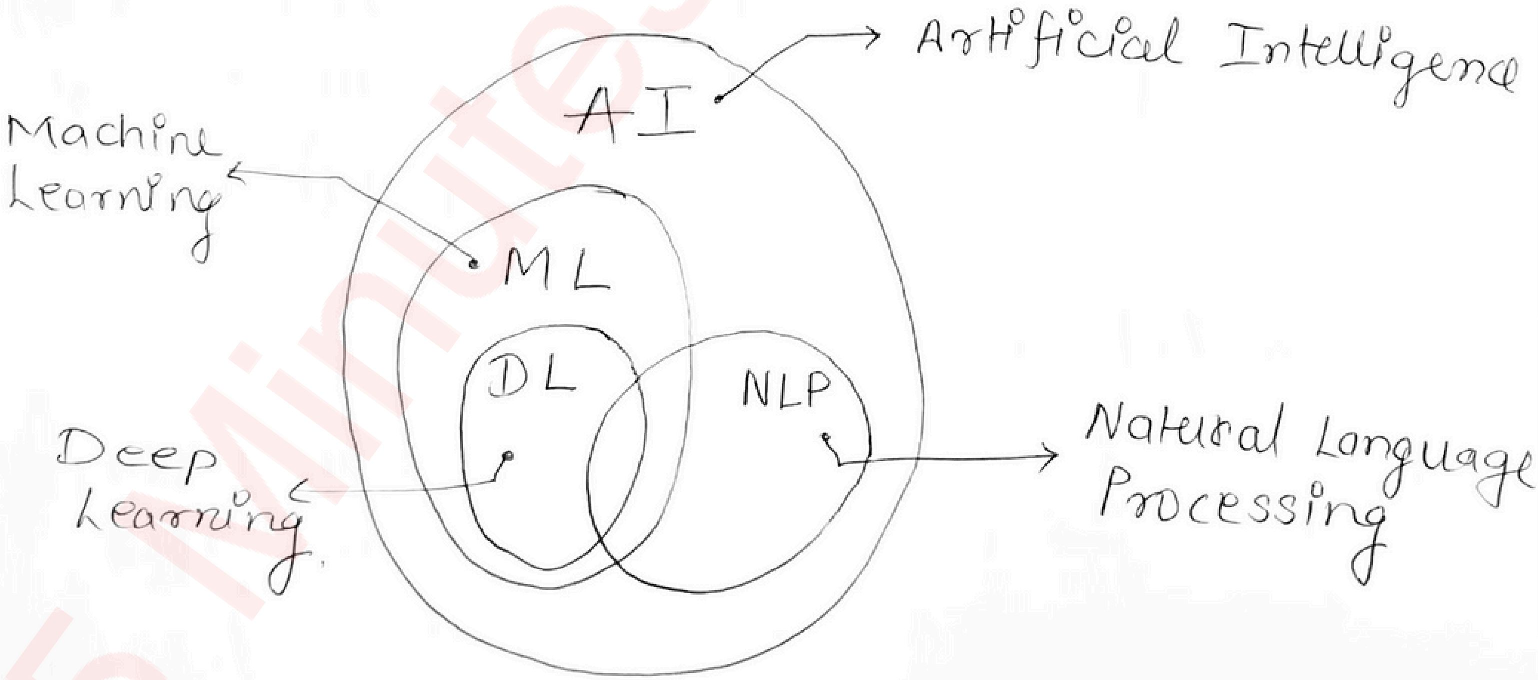
Artificial   Intelligence

→ Intelligence exhibited by machines

Intelligence → Human

machines → Computer Systems

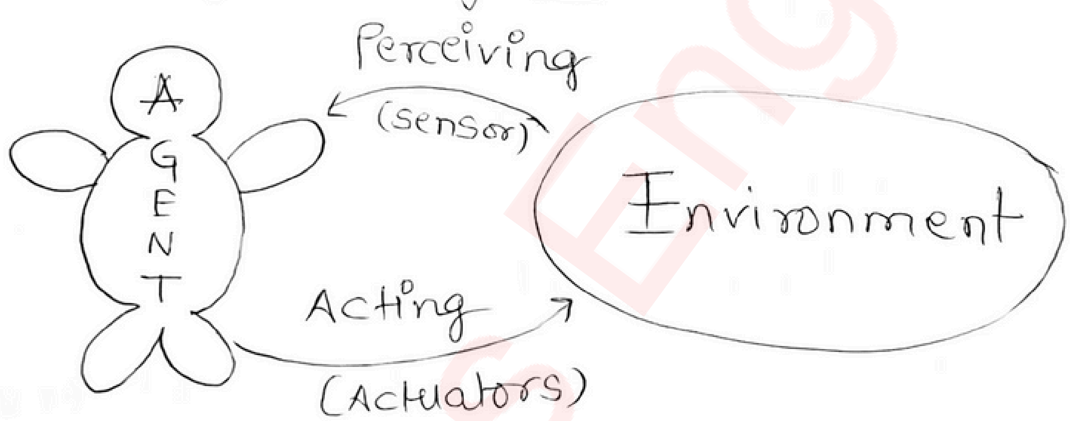
- Reasoning
- Learning
- Problem solving
- Perception.



- Types of AI
  - Purely Reactive
    - NO memories, NO data or past experience
    - focus on current scenarios & React
    - Eg: Google's AlphaGO.
  - Limited Memory
    - Can store past experiences or data for short time / limited period.
    - Eg: self-driving cars.
  - Theory of Mind
    - Understand human emotion & beliefs
    - Able to interact socially like us.
    - Yet to be built.
    - eg: Sofia.
  - Self-Aware
    - future of AI (super intelligent)
    - own sentiment & conscious.
    - SA machines ≥ humans [Chitti]
    - Hypothetical concept.

- Future of AI
  - → Education
  - → Healthcare
  - → Manufacturing
  - → Entertainment
  - → Defence

- Intelligent Agents



1. AI agent must be able to perceive the enviroment.
2. Make decision based on sensed data.
3. Action must be generated & performed based on decision.

* High performance
* Best/optimized Result
* Rational Action

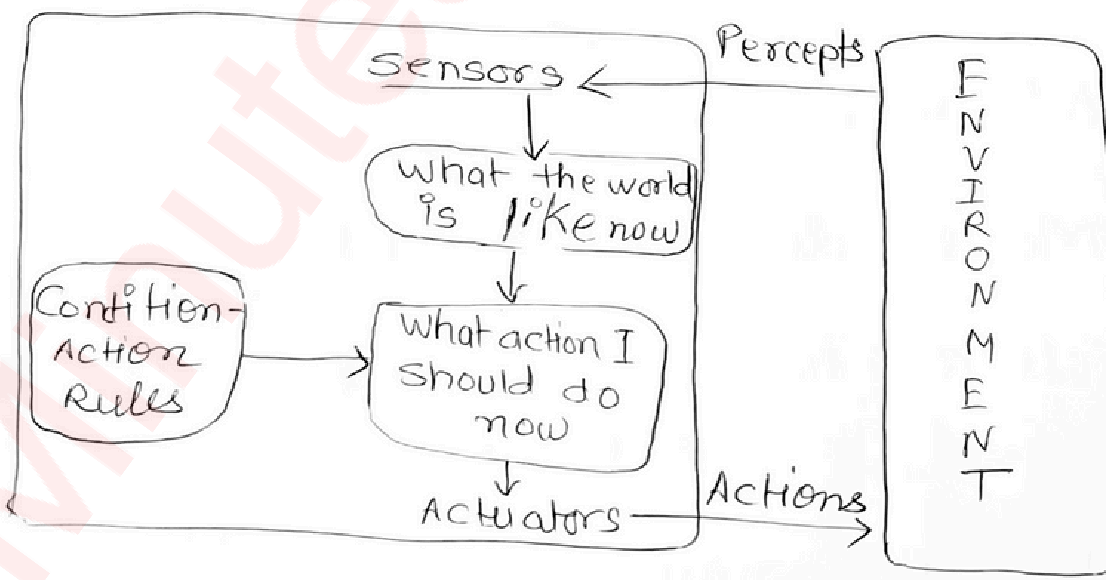
} Goals of Agent.

Types:

- → Simple Reflex Agents
- → Model Based Agents
- → Goal Based Agents
- → Utility Based Agents
- → Learning agents.

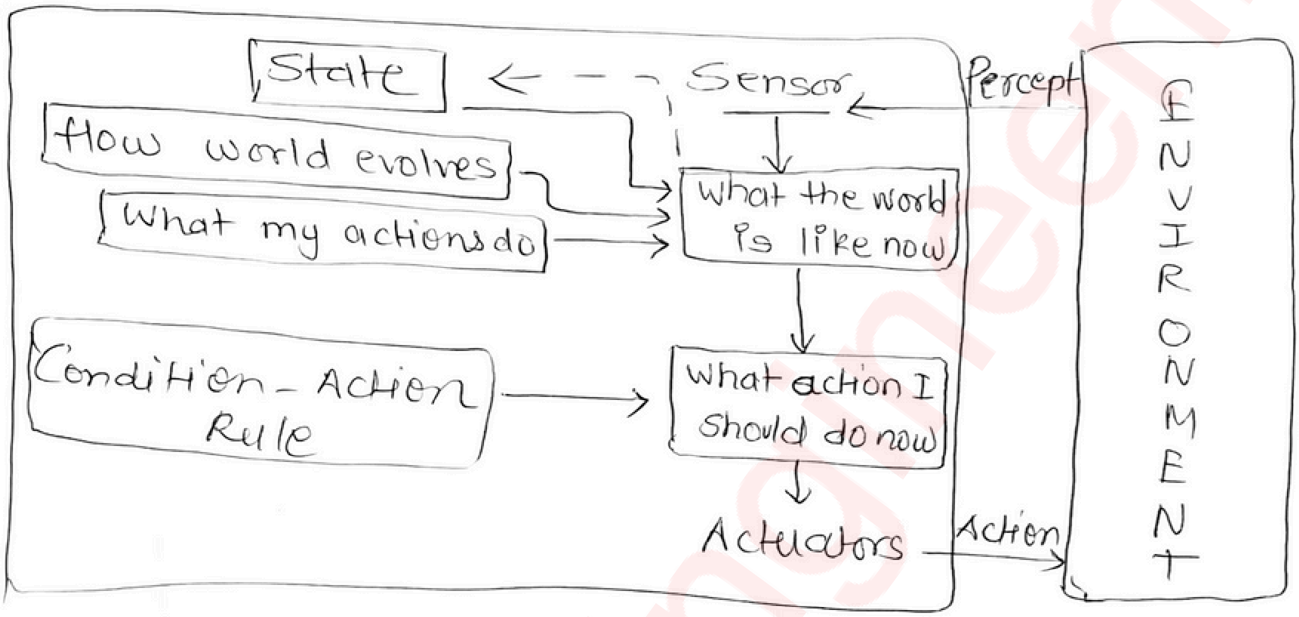
① Simplex Reflex Agents

- → Current perception
- → Based on If-then Rules
- → Condition-action Rule
- → very limited intelligence
- → Operates on fully observable environment.



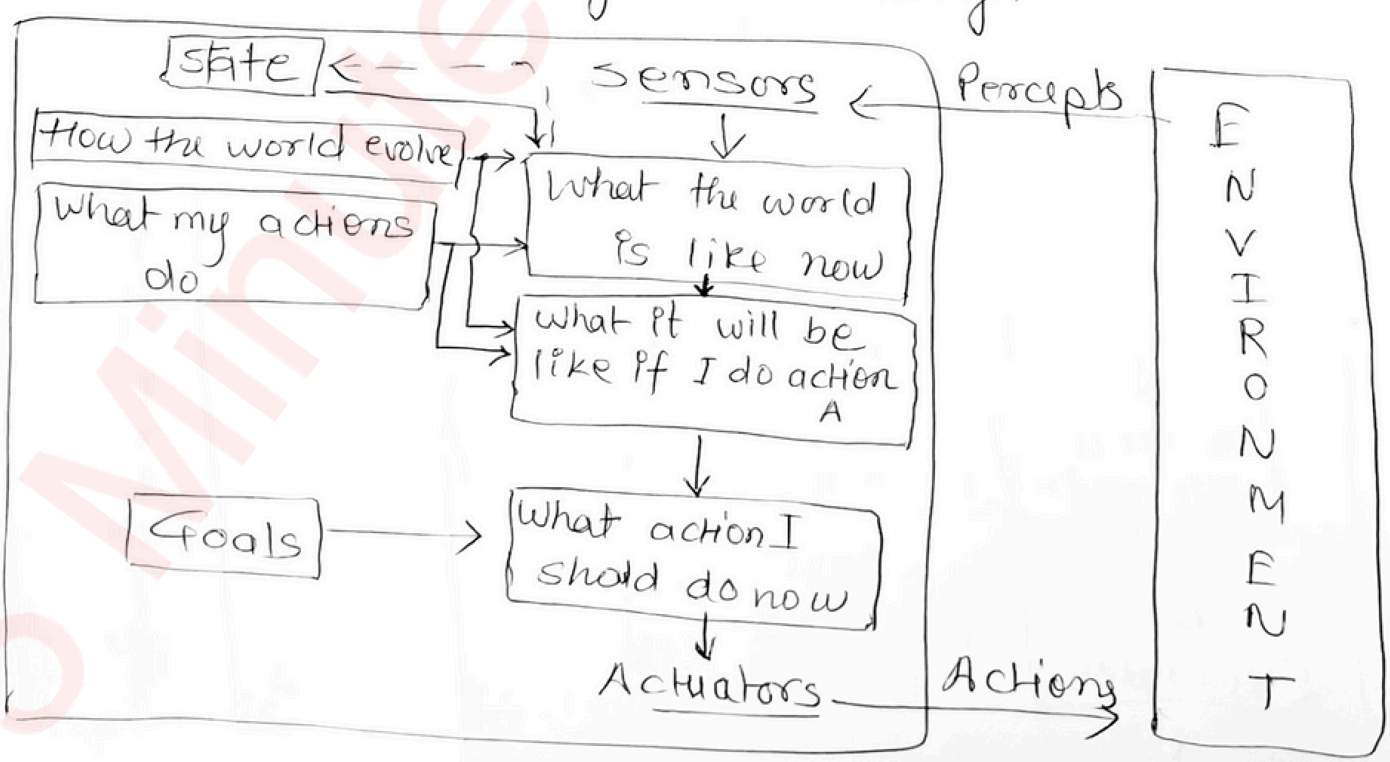
② Model Based Agents

- → Partially observable environment.
- → Internal state /store percept history.



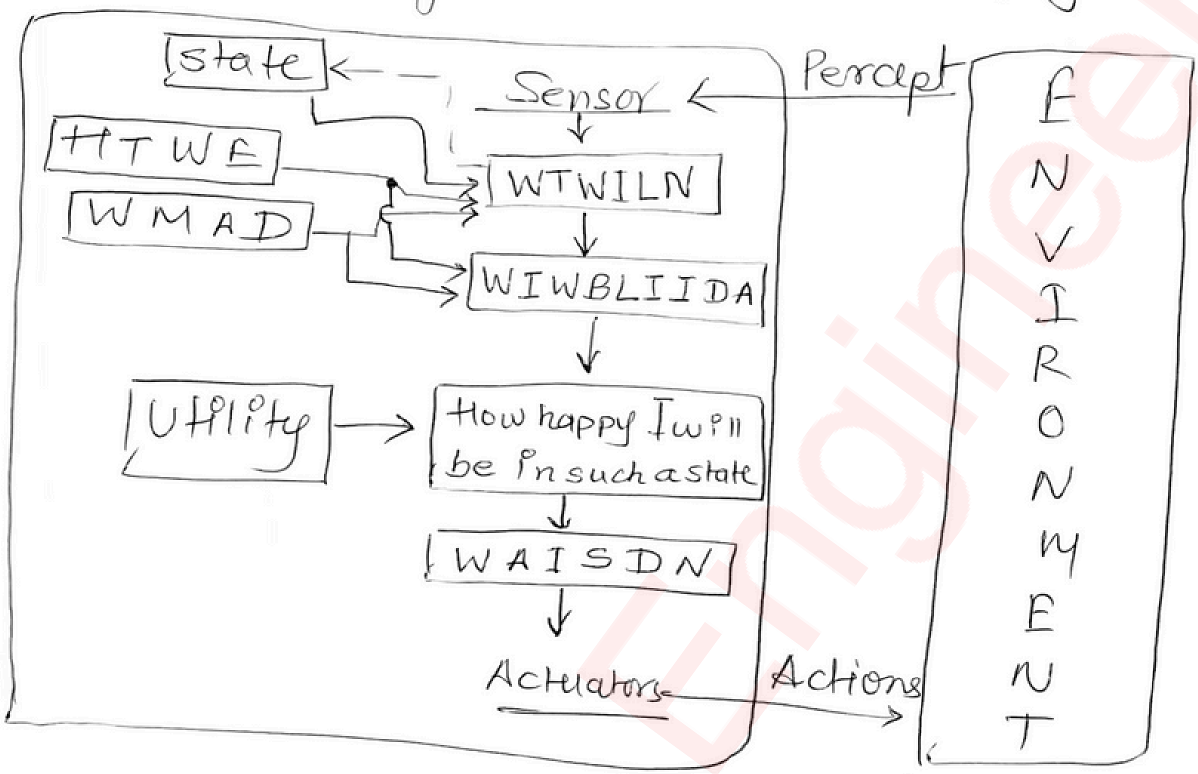
③ Goal Based Agents

- → Goal: Description of desirable situation.
- → Searching & Planning.



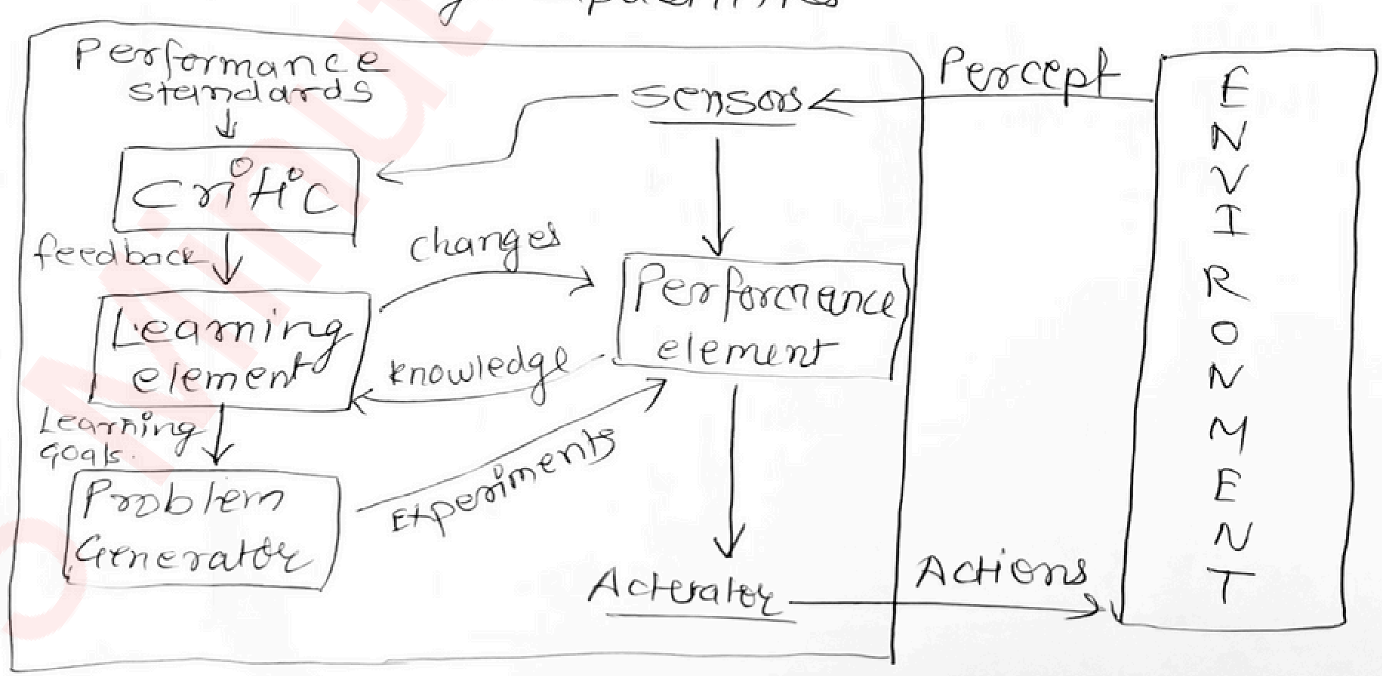
④ Utility Based Agents

- → Multiple option, decide/go for best.
- → Choose actions are based on utility.
- → Utility describes how happy the agent is.



⑤ Learning Agent

- → Learn from its past experiences.
- → Learning capabilities.



- Environments
  - → Fully Vs Partially Observable.
  - → Static Vs Dynamic
  - → Discrete Vs Continuous
  - → Episodic Vs Sequentials
  - → Known Vs Unknown
  - → Accessible Vs Inaccessible
  - → Single Agent Vs Multi Agents

- Fuzzy Logic
  - → father of FL : Lotfi Zadeh
  - → Mathematical language

* Boolean Logic * Relational Logic * Predicate Logic

↓

fuzzy Logic

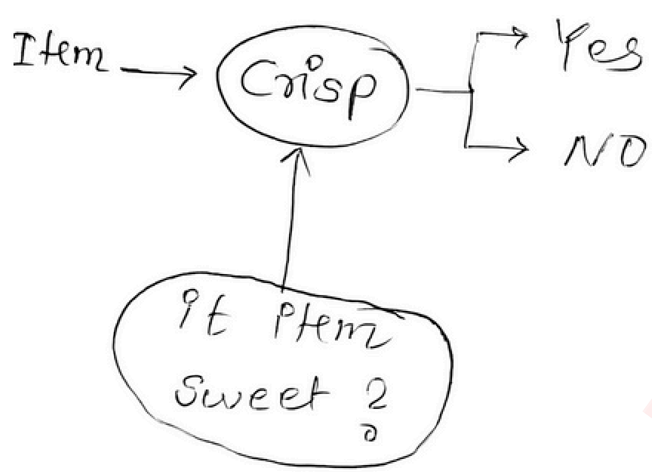
↓

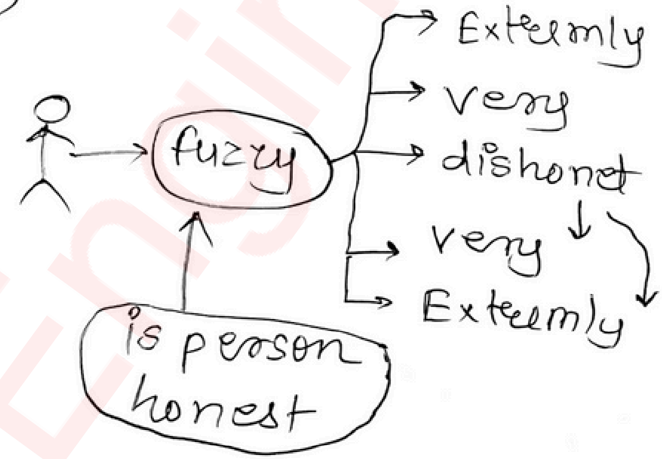
deals with fuzzy set / fuzzy Algebra

- Crisp Logic Vs fuzzy Logic

Crisp o/p
- → Yes / NO
- → True / false

fuzzy o/p
- → May be
- → May not be
- → Absolutely



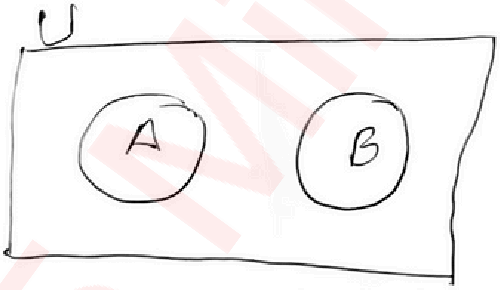


"uncertainty"

- Crisp set

↓

U : All students
A : 10th
B : 12th



- fuzzy sets

↓

U : All students
G : Good students
B : Bad students

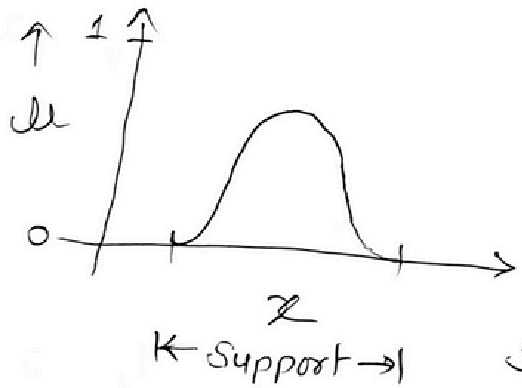
$G = \{G, \mu(G)\}$ → degree (MF)

$G = \{(A, 0.9), (B, 0.7)\}$

$B = \{(A, 0.1), (B, 0.3)\}$

- Membership function features:

① Support : set of all points, such that $\{\mu_A(x) > 0\}$

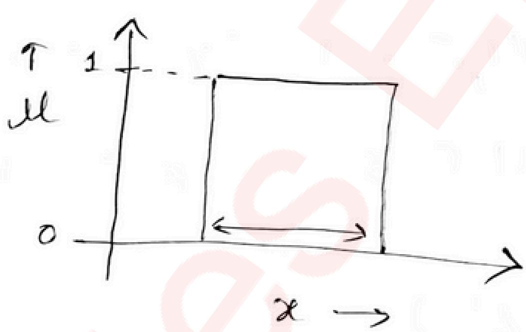


$$Support(A) = \{x \mid \mu_A(x) > 0\}$$

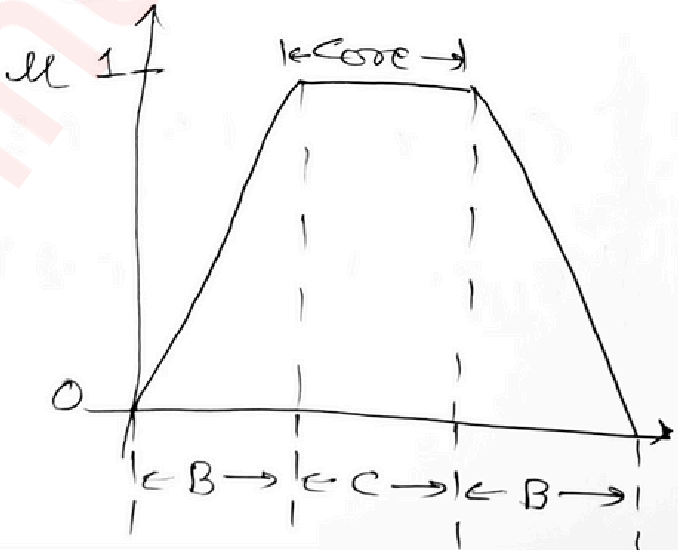
② Core : $\mu_A(x) = 1$

$$Core(A) = \{ x = \mu_A(x) = 1 \}$$

③



③ Boundary : $(1 > \mu_A(x) > 0)$



Fuzzy Logic Operations:

① Union : $(A \cup B)$

$$\mu_{(A \cup B)}(x) = \max(\mu_A(x), \mu_B(x))$$

② Intersection: $(A \cap B)$

$$\mu_{(A \cap B)}(x) = \min(\mu_A(x), \mu_B(x))$$

example: $A = \{(x_1, 0.6), (x_2, 0.7), (x_3, 0.4)\}$
$B = \{(x_1, 0.3), (x_2, 0.2), (x_3, 0.5)\}$

$(A \cup B) = \{(x_1, 0.6), (x_2, 0.7), (x_3, 0.5)\}$
$(A \cap B) = \{(x_1, 0.3), (x_2, 0.2), (x_3, 0.4)\}$

③ Complement (A^C)

$$\mu_{(A^C)}(x) = 1 - \mu_A(x)$$

$A^C = \{(x_1, 0.4), (x_2, 0.3), (x_3, 0.6)\}$
$B^C = \{(x_1, 0.7), (x_2, 0.8), (x_3, 0.5)\}$

④ Vector Product $(A \cdot B)$

$$\mu_{(A \cdot B)}(x) = \mu_A(x) \cdot \mu_B(x)$$

$$A \cdot B = \{(x_1, 0.18), (x_2, 0.14), (x_3, 0.20)\}$$

⑤ Scalar Product $(\alpha \times A)$

$$\mu_{\alpha A}(x) = \alpha \times \mu_A(x)$$

⑥ Equality $(A = B)$

$$\mu_A(x) = \mu_B(x)$$

⑦ Power (A^α)

$$\mu_{(A^\alpha)}(x) = (\mu_A(x))^\alpha$$

⑧ Sum $(A + B)$

$$\mu_{A+B}(x) = \mu_A(x) + \mu_B(x) - \mu_A(x) \cdot \mu_B(x)$$

⑨ Difference $(A - B)$

$$\mu_{A-B}(x) = \mu_{A \cap B^c}(x)$$

⑩ Disjunctive Sum $(A \oplus B)$

$$\mu_{(A \oplus B)}(x) = (A^c \cap B) \cup (A \cap B^c)$$

- fuzzy set Properties

① Commutative :- $A \cap B = B \cap A$
$A \cup B = B \cup A$

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

③ Distributive :- $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$
$A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$

④ Idempotence : $A \cup A = A$; $A \cap A = A$
$A \cup \phi = A$; $A \cap \phi = \phi$

⑤ Transitive : If $A \subseteq B$; $B \subseteq C$ then $A \subseteq C$

⑥ Demorgan's Law :- $(A \cap B)^C = A^C \cup B^C$
$(A \cup B)^C = A^C \cap B^C$

- fuzzy Relation

$A = \{(x_1, 0.6), (x_2, 0.2), (x_3, 0.3)\}$
$B = \{(y_1, 0.7), (y_2, 0.3), (y_3, 0.4)\}$

$\mu_R(x,y) = \mu_{A \times B}(x,y) = \min(\mu_A(x), \mu_B(y))$

$$R = A \times B = \begin{array}{c} \\ X1 \\ X2 \\ X3 \end{array} \begin{array}{c} \begin{array}{ccc} Y1 & Y2 & Y3 \end{array} \\ \begin{bmatrix} 0.6 & 0.3 & 0.4 \\ 0.2 & 0.2 & 0.2 \\ 0.3 & 0.3 & 0.3 \end{bmatrix} \end{array}$$

Fuzzy if then Rule

$T_{High} = \{(25, 0.1), (30, 0.2), (35, 0.5), (40, 0.6)\}$

$P_{Low} = \{(2, 0.3), (5, 0.5), (6, 0.4)\}$

If Temp is high then pressure is low

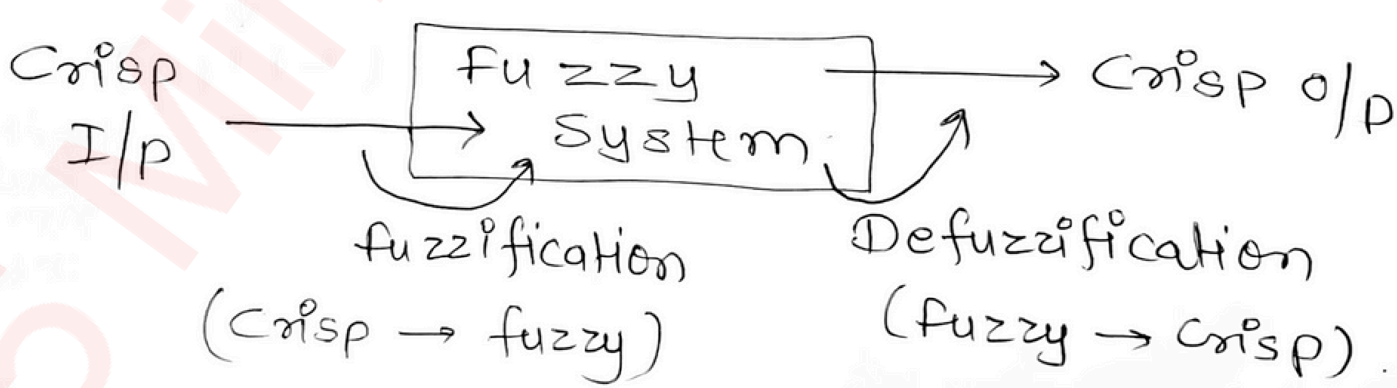
$$R: A \longrightarrow B \qquad (A \times B) \rightarrow R$$

A → premise

B → Conclusion

$$R: T_{high} \longrightarrow P_{Low}$$

$$R = \begin{array}{c|c} & \begin{array}{ccc} 2 & 5 & 6 \end{array} \\ \begin{array}{c} 25 \\ 30 \\ 35 \\ 40 \end{array} & \begin{bmatrix} 0.1 & 0.1 & 0.1 \\ 0.2 & 0.2 & 0.2 \\ 0.3 & 0.5 & 0.4 \\ 0.3 & 0.5 & 0.4 \end{bmatrix} \end{array}$$

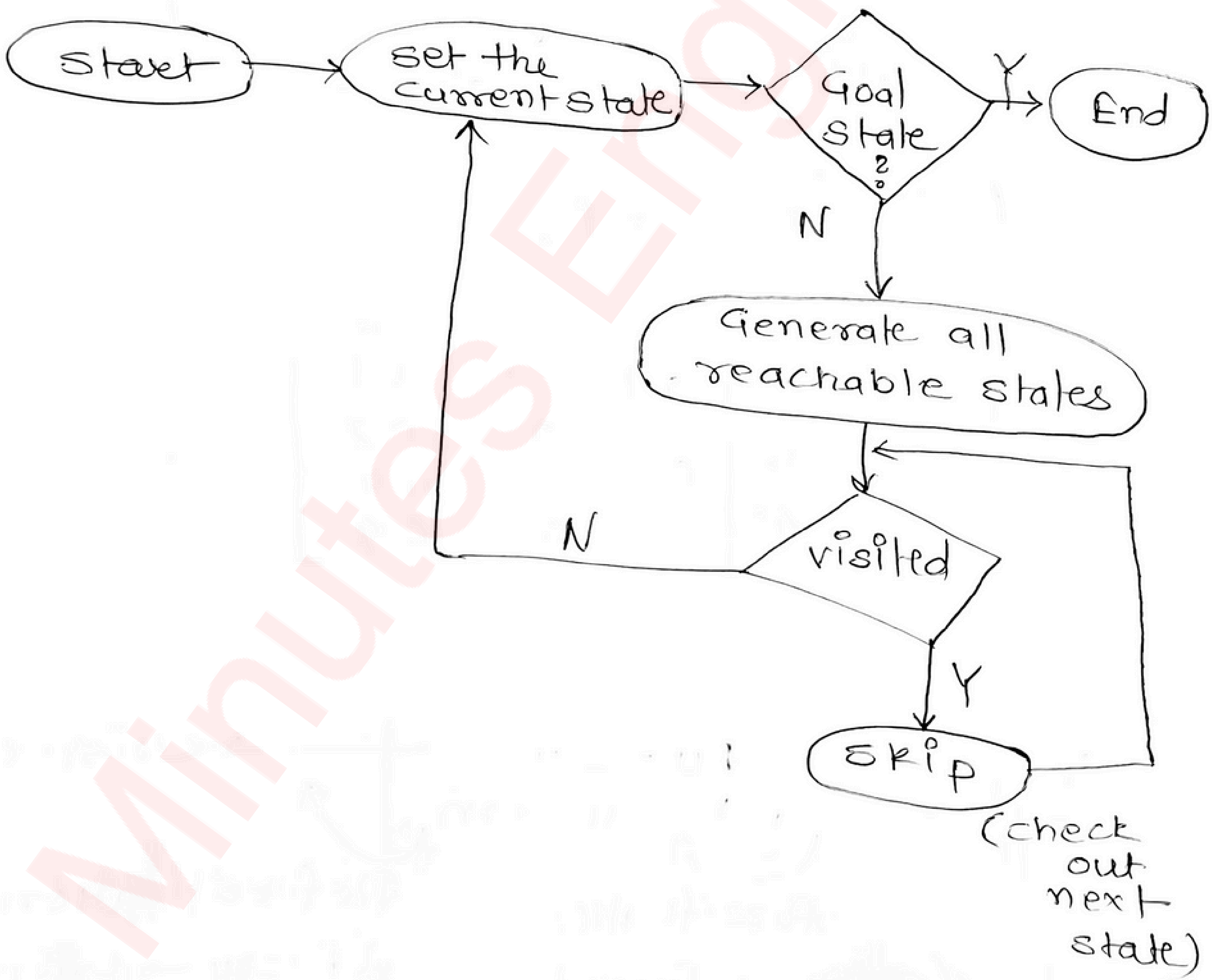


* Problem solving: Solve the given problem
(But not "you" ☺)
[huge large problems]

→ State space search:

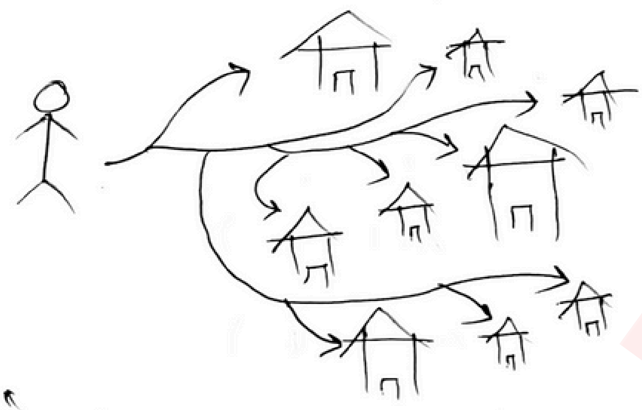
↳ S : { S, A, Action(s), Result(s, a), Cost(s, a) }

- Exhaustiveness
- Completeness
- Optimality



Uninformed search Vs Informed search

Uninformed search	Informed search
① No Info (Blind search)	① with info (Heuristic) search
② Time consuming	② Fast & Quick
③ No knowledge	③ use knowledge to reach Goal.
④ More complex (T&S)	④ Less complex (T&S)
⑤ Eg: BFS, DFS.	⑤ A*, heuristic DFS, Best first search, AO*



" visit & knok"

" Ask someone".

⇒ "May/ May be best/optimal soln"

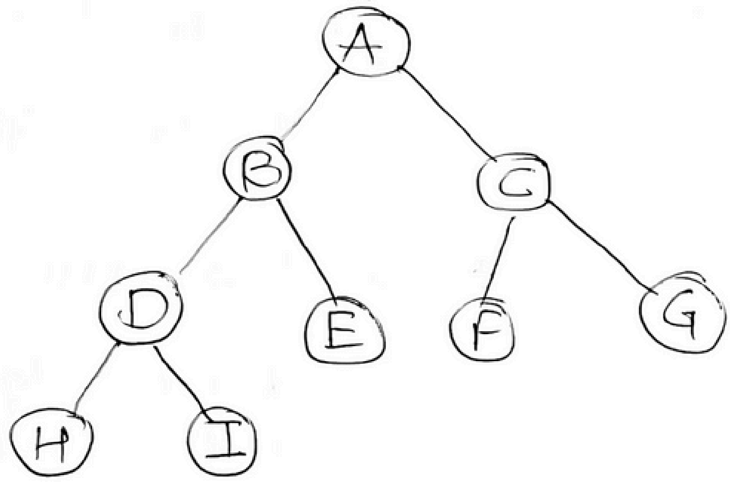
⇒ "correct/wrong addr".

- Uninformed searching

① BFS (Breadth first search)

- → FIFO (Queue)
- → Shallowest Node
- → Complete.
- → Optimal
- → Level wise.

Eg:



Time Complexity

↓

$O(b^d)$

b: branch

d: depth

$\Rightarrow 2^3 = 8$ max

$\Rightarrow$ ~~A~~ Queue

B C ← (After A)

C D E ← (After B)

D E F G ← (After C)

E F G H I ← (After D)

F G H I ← (After E)

G H I ← (After F)

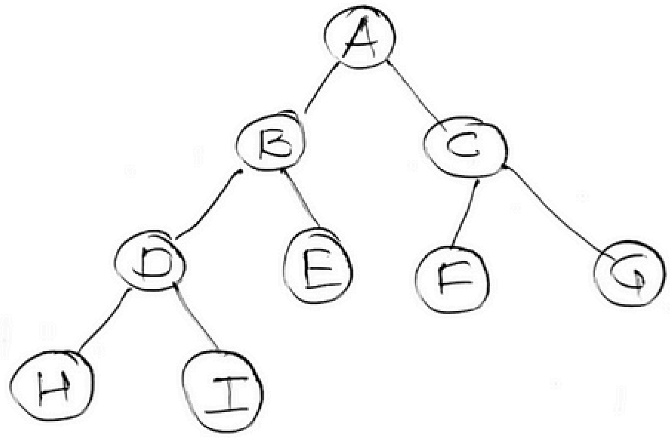
H I ← (After G)

I ← (After H)

② DFS

- → Depth first search
- → LIFO (Stack)
- → Deepest Node
- → Incomplete
- → Non optimal

Eg.:

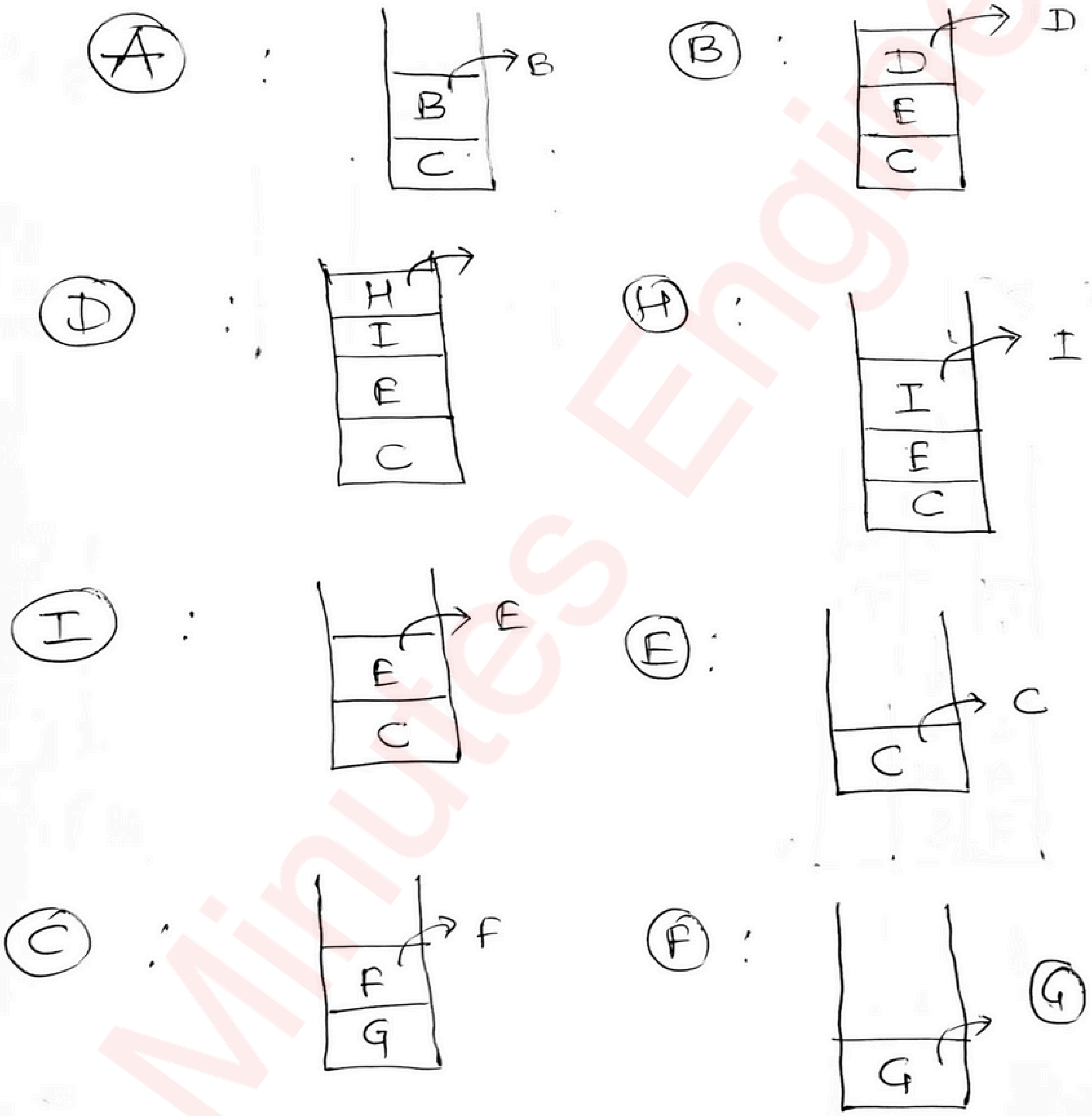


Time Complexity

↓

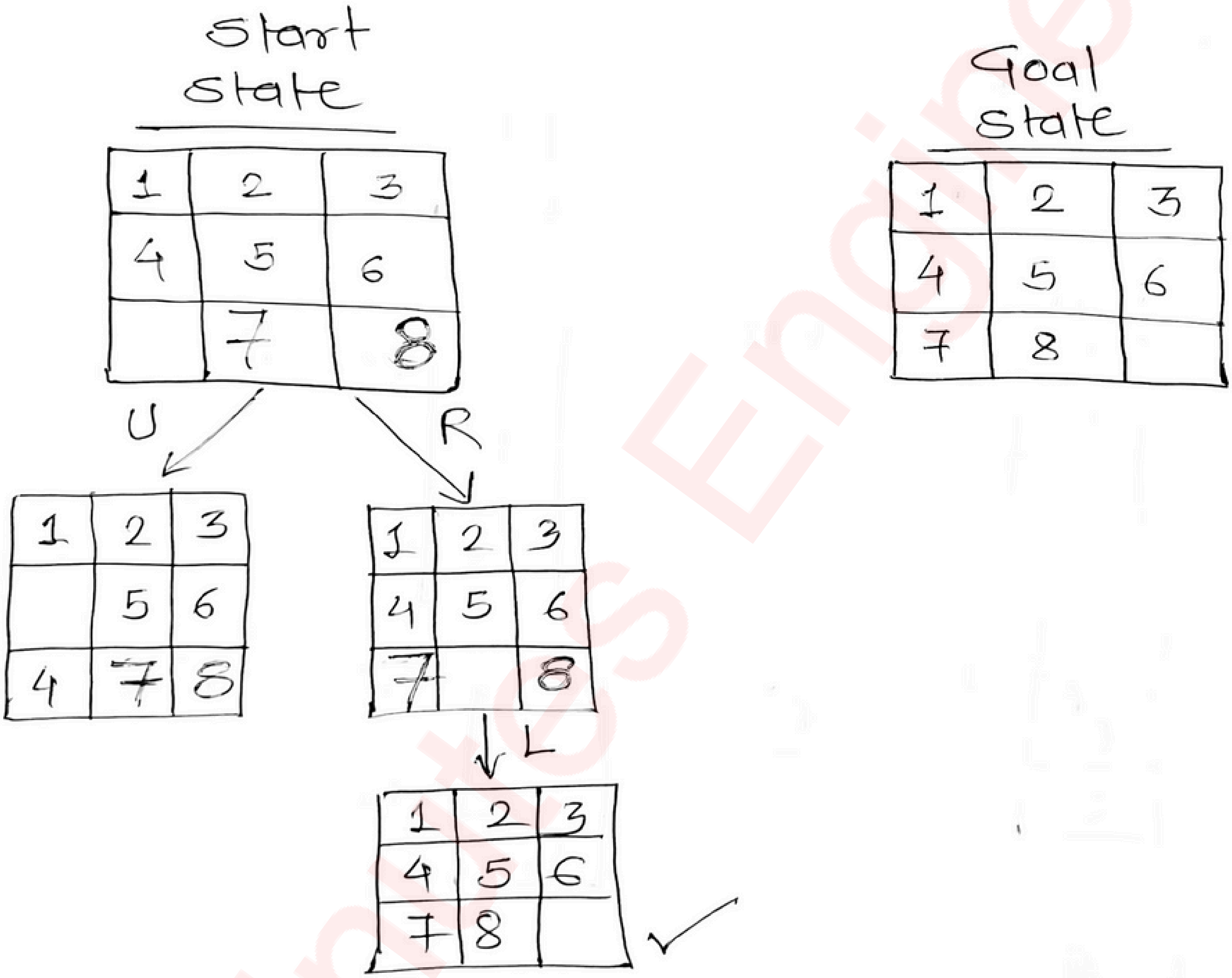
$O(b^d)$

⇒ A B D H I E C F G



8 Puzzle Problem [without Heuristic]

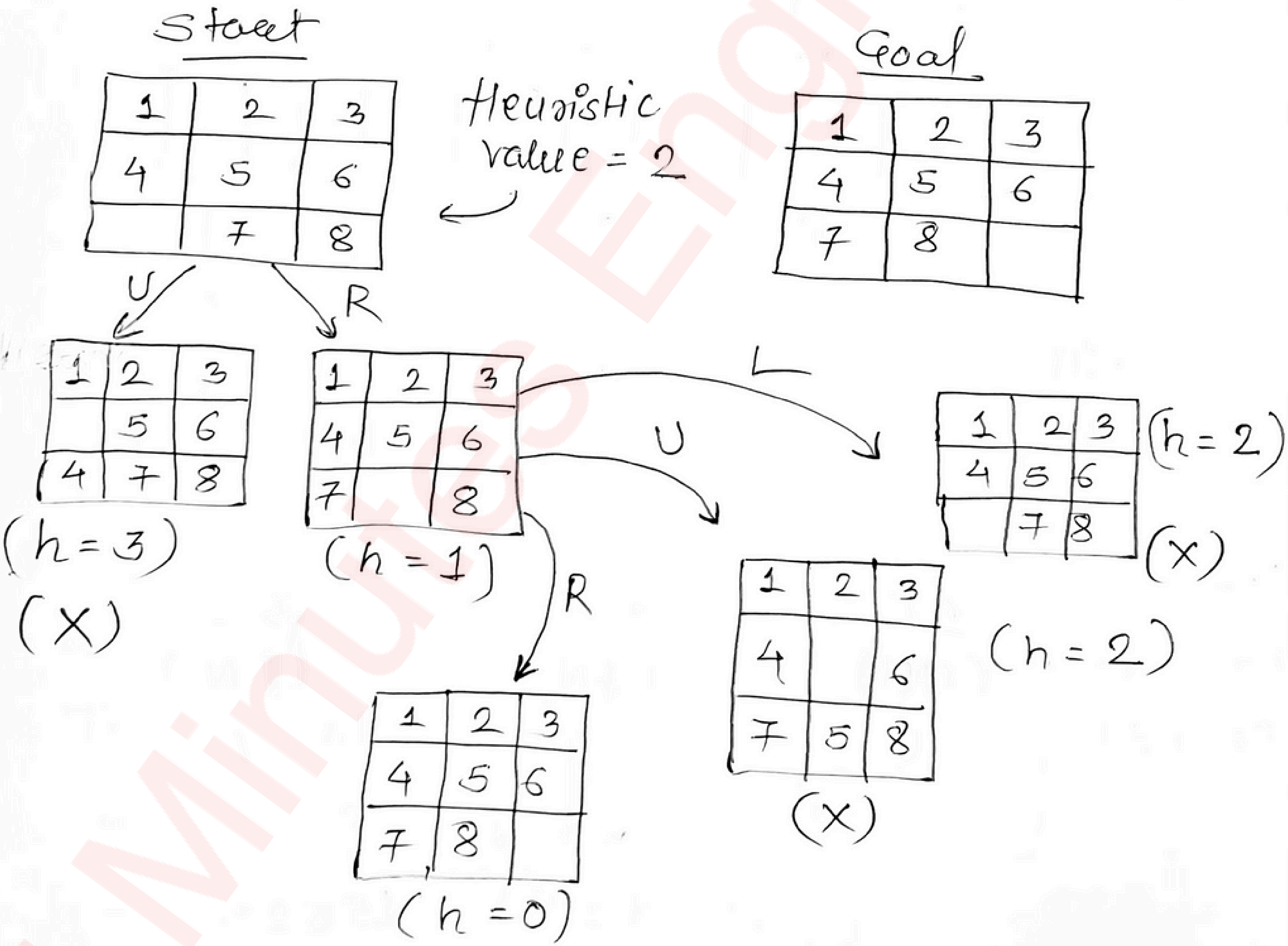
- → 4 moves (Up, down, Left Right)
- → BFS [$O(b^d)$]



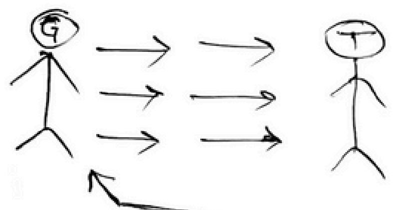
• Heuristic

- Used to estimate the cost of reaching a goal state from a given state.
- No exhaustive exploring.
- Quick & fast way to reach a sol^n.

• 8 Puzzle Problem (with Heuristic)



- Generate & Test
  - → Heuristic
  - → DFS with backtracking



→ Accept (If S = G)

→ skip/drop (If S ≠ G)

→ Complete

→ Non-Redundant

→ Informed.

Eg:

(0 – 99)		(0 – 99)		(0 – 99)	
0	0	0	0	0	1
					2
					3
					|
					|

Non-heuristic: $(100)^3 \rightarrow$ 1M

Heuristic (No.s are prime): 25 (PNs), 25 (PNs), 25 (PNs)

$(25)^3 = 15625$

- Best first search

Algorithm:

Let OPEN be a 'P' queue containing start state

Loop

If OPEN is empty return fail.

Node ← Remove first (OPEN)

If Node is Goal

the return path from start to node

else generate all successors of Node & put the newly generated Node in OPEN according to their 'h' values.

END LOOP.

→ Not Complete

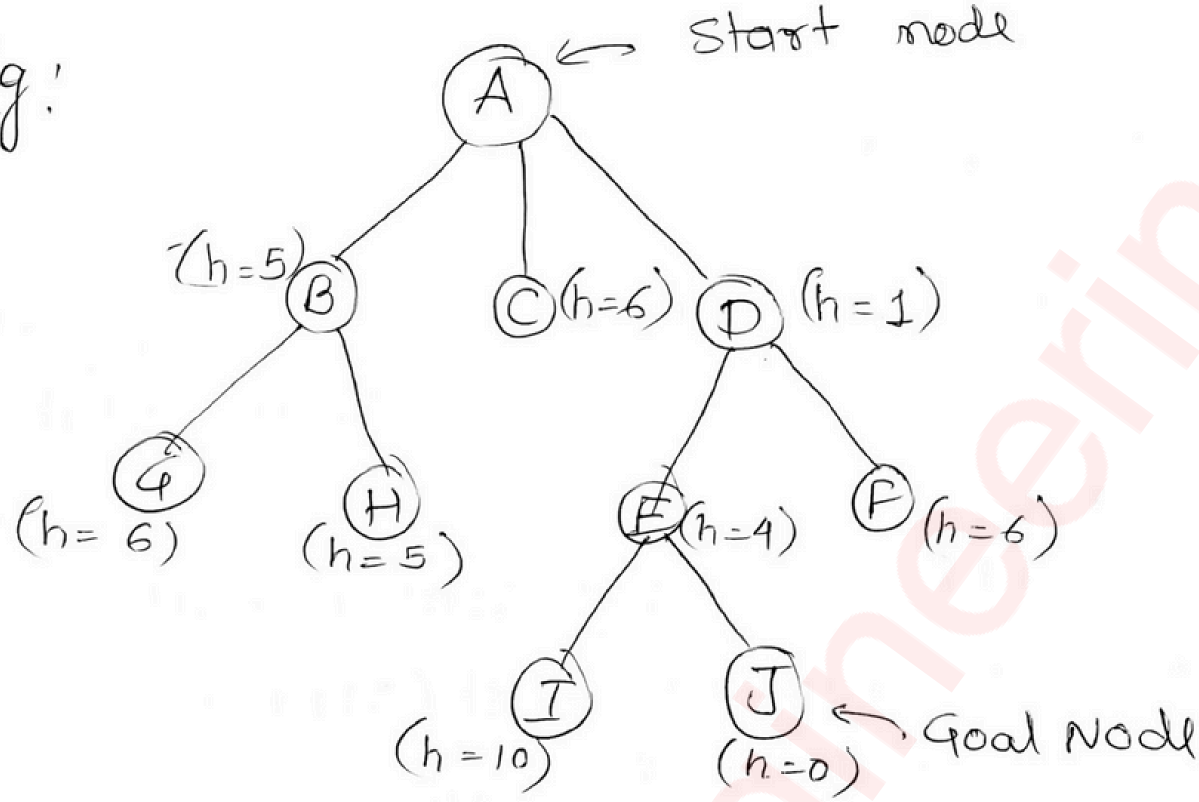
→ Optimal

→ Time Complexity = O(bm)

→ Space Complexity = O(bm)

→ h = euclidean distance = $\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$

Eg:



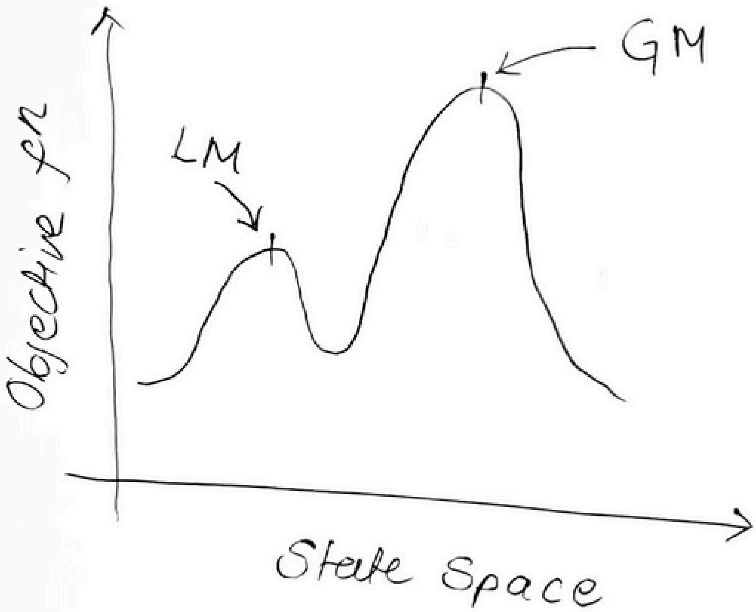
⇒ Open Queue (Priority) :

D	B	C			}	A
E	B	C	F		}	D
J	B	C	F	I	}	E
0	5	6	6	10		↓
						J

final soln path

$\boxed{A \rightarrow D \rightarrow E \rightarrow J}$

° Hill climbing Algorithm



Not → Complete
→ Optimal.

° Problems:

→ Local Maxima
→ Plateau

→ Ridge

I/p → Problem

Local variables → current & neighbor

current ← make_node (Initial state (problem))

Loop
  do
    neighbor ← A highest valued successor of current.
    If value (neighbor) ≤ Value (current)
    then
        return state (current)
    else
    current ← neighbor .

A* Algorithm

- Informed searching
- $f(n) = g(n) + h(n)$

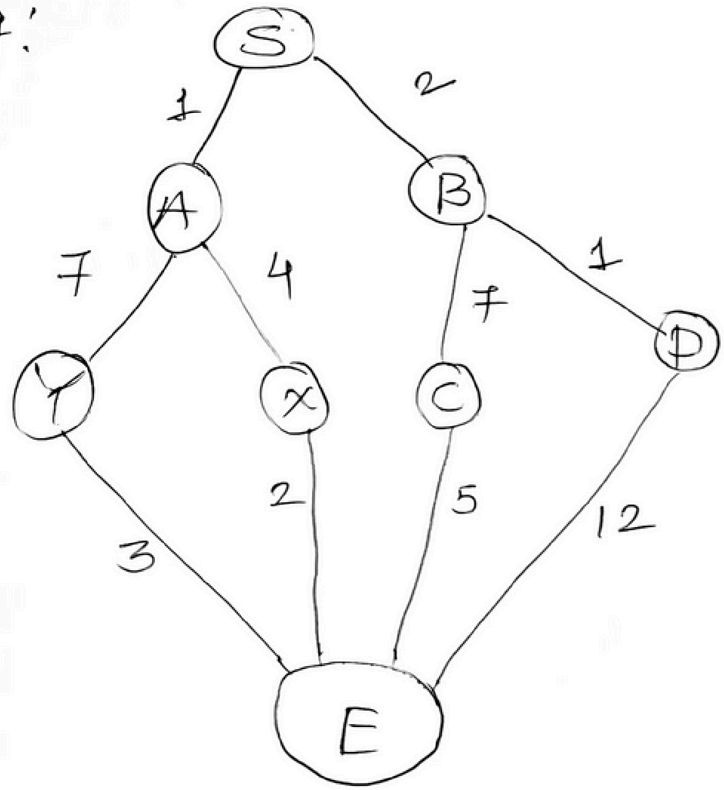
$f(n)$ → Evaluation fn.

$g(n)$ → How much You travelled from initial node

$h(n)$ → heuristic (Estimation cost from 'i' to 'G')

$$\underbrace{S \longrightarrow i}_{g(n)} \underbrace{\longrightarrow G}_{h(n)}$$

Eg:



$A \rightarrow 5$

$B \rightarrow 6$

$C \rightarrow 4$

$D \rightarrow 15$

$x \rightarrow 5$

$Y \rightarrow 8$

Yes { → Complete
→ optimal.

① Expand (S):

(S, A) $f = 1+5 = 6$; (S, B) $f = 2+6 = 8$

② Expand (A):

(S, B) $f = 2+6 = 8$; (S, A, x) $f = (1+4)+5 = 10$

(S, A, Y) $f = (1+7)+8 = 16$

③ Expand B :

(S, A, x) $f = (1+4)+5 = 10$; (S, A, Y) $f = (1+7)+8 = 16$

(S, B, C) $f = (2+7)+4 = 13$; (S, B, D) $f = (2+1)+15 = 18$

④ Expand x :

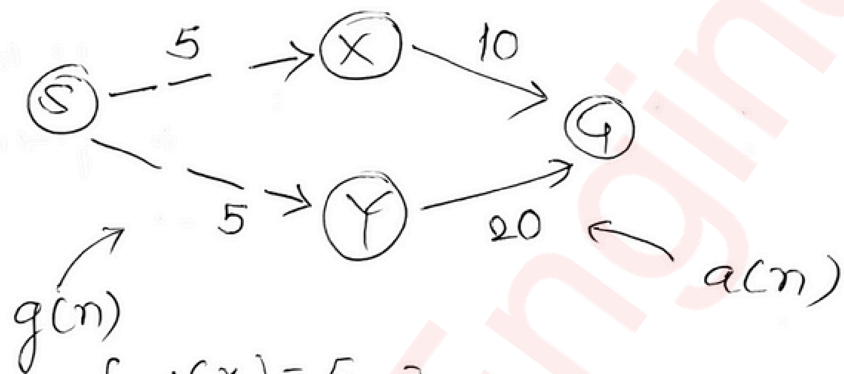
(S, A, x, E) is the best path with actual cost = 7.

- Overestimation & Underestimation

$e(n) \geq a(n)$ (overestimation) $\qquad$ $e(n) \leq a(n)$ (underestimation)

$e()$: estimated
$a()$: actual.

eg:-



$$\left\{ \begin{array}{l} g(x) = 5 \\ g(y) = 5 \end{array} \right\}$$

Case I: overestimation

$h(n) = e(n) = $ estimated.

$h(x) = e(x) = 50$

$h(Y) = e(Y) = 40$

$\therefore f(X) = g(x) + h(x) = 5 + 50 = 55$

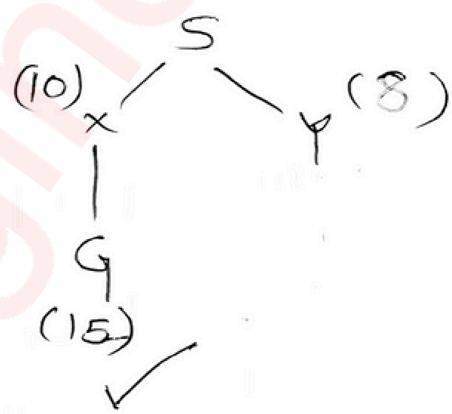
$f(Y) = g(Y) + h(Y) = 5 + 40 = 45$ ✓

$f(G) = g(G) + h(G)$
$= 25 + 0 = 25$ (actual cost)

Case 2: underestimation

$$h(X) = e(X) = 5$$
$$h(Y) = e(Y) = 8$$

$$f(X) = g(X) + h(X) = 5 + 5 = 10 \checkmark$$
$$f(Y) = g(Y) + h(Y) = 5 + 8 = 13$$

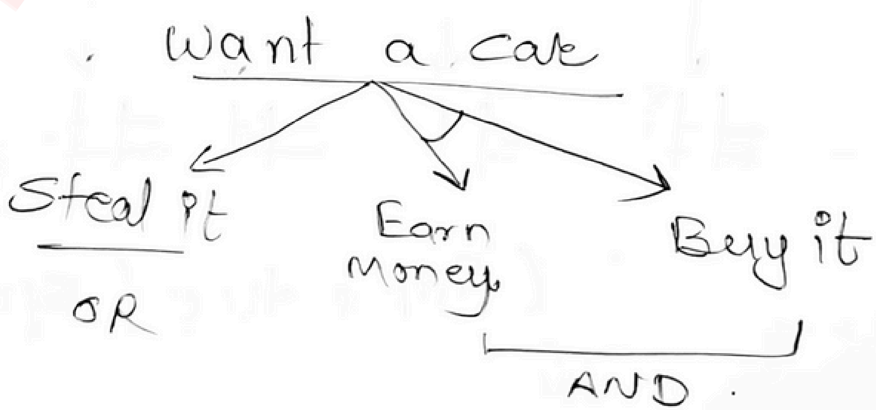


from (Y) $f(G) = g(G) + h(G)$
$= 25 + 0$
$= 25$

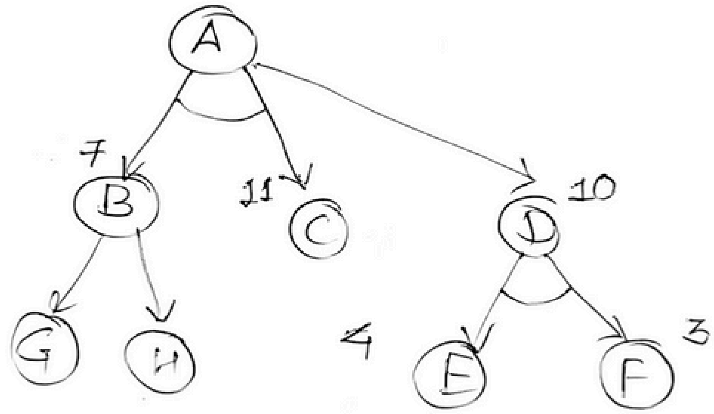
from (X) $f(G) = g(G) + h(G)$
$= 15 + 0$
$= 15 \checkmark$

- **AO* Algorithm**:
  - → AND/OR
  - → Problem Decomposition

eg:-



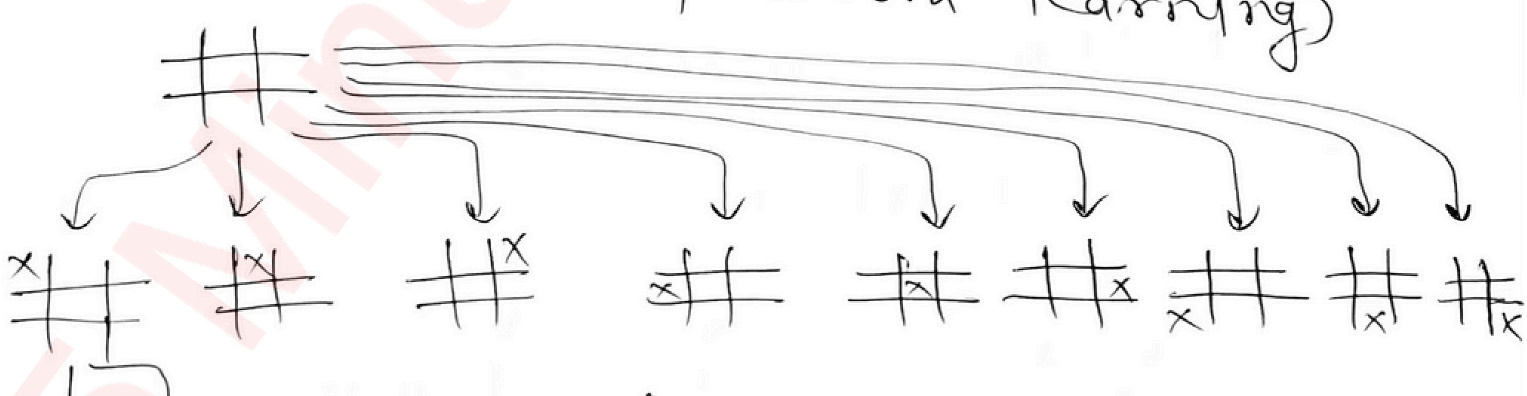
→ eg:



→ Doesn't explore all paths.

* Game playing
  - → Legal moves
  - → win / lose conditions of game.

Types
  - → Rule-Based (Predefined Rules) (Tic-Tac-Toe)
  - → Machin Learning - Based. (Reinforcement learning)



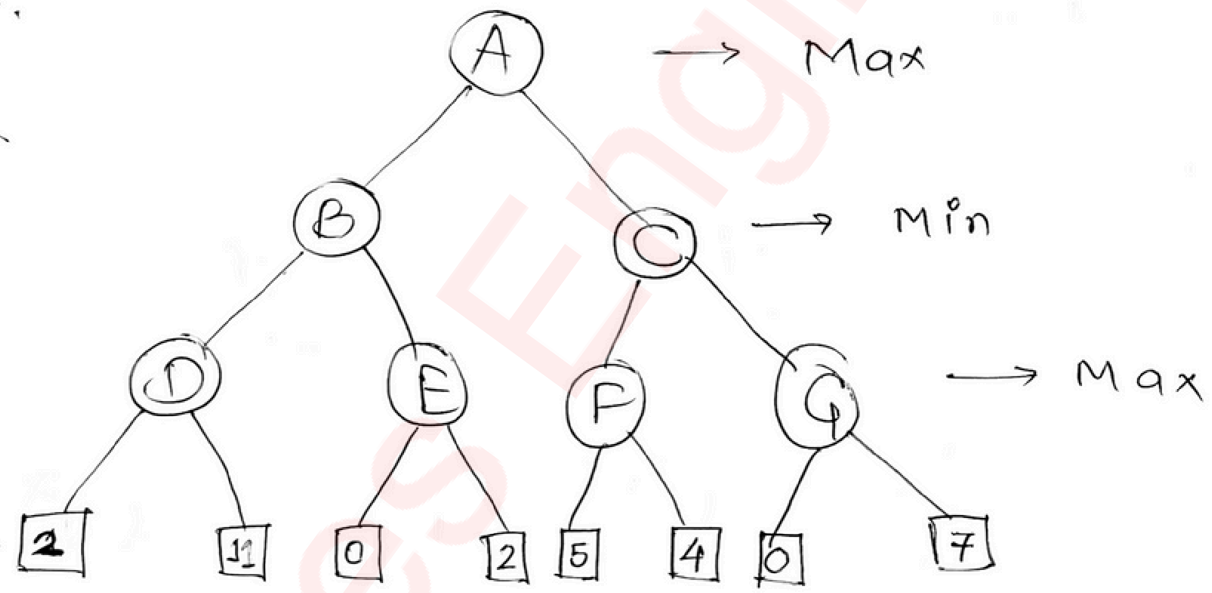
(Game tree / Space Graph)

* Minimax Algorithm :

- → Backtracking
- → Max → Player 1
- → Min → Player 2
- → Max select maximized value (Best move)
- → Min select minimized value. (Worst move) for Max.

Eg. :-

Game tree



TC : $O(b^d)$

SC : $O(bd)$ ≈ DFS

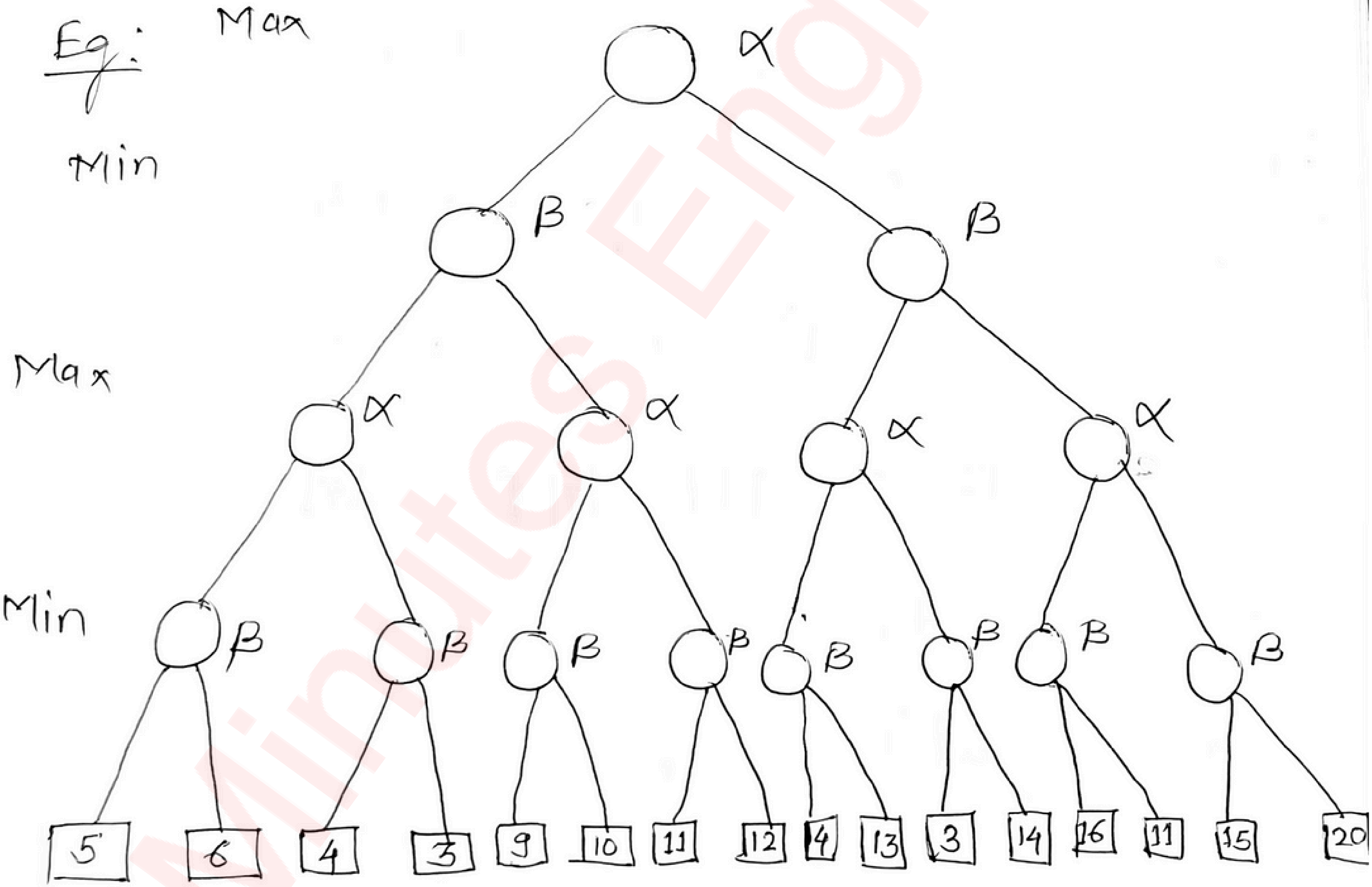
* Alpha Beta Pruning
(α) (β)

↳ Prerequisites : Minimax algorithm

α → Best choice (highest utility/value) [maximizer]

β → Best choice (lowest utility/value) [minimizer]

Eg:



* Knowledge Representation
  - → Declarative (objects & facts)
  - → Structural (Objects & Relationship)
  - → Procedural (Rules)
  - → Heuristic (estimate, previous/history/experiences) expert knowledge.
  - → Meta (K about K)

- Techniques
  - → Logical
  - → semantic Net
  - → Frame
  - → Production Rules.

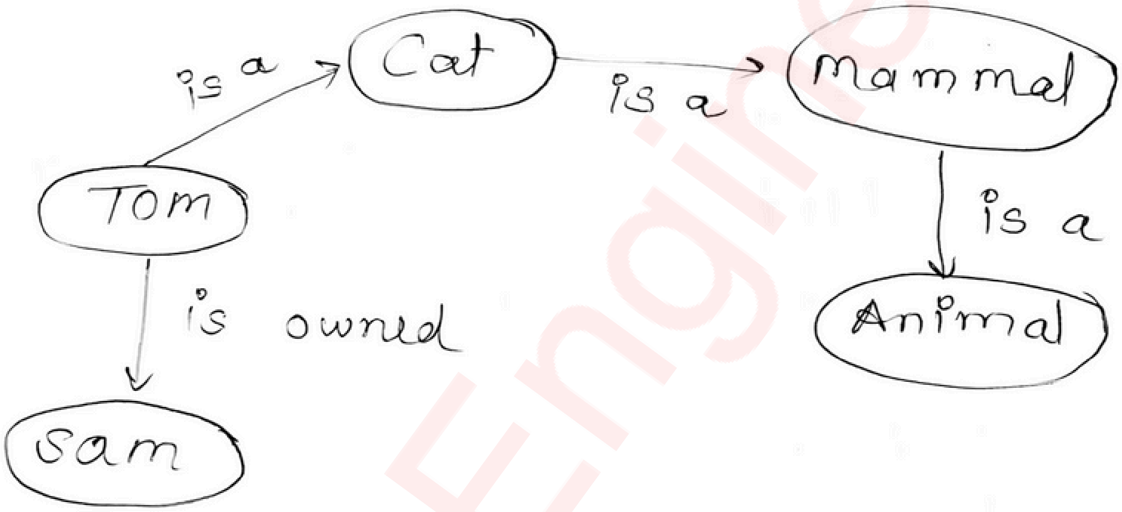
1. Logical Representation
   - → Propositional
   - → Predicate

② Semantic Network

→ "is a"
→ "kind of"
→ is owned

} Relation.

eg:-



③ Frame

→ Slots (objects/entity).
→ fillers (Attributes)
→ eg: "Ticket" { movie, travel, event }

④ Production Rules

→ If-Then
→ eg: If (I watch 5ME) then (I can easily write exam).

Propositional Logic: (True or False) (Not Both) at the same time.

eg:- 2 + 3 = 5 (True) ✓

watching AI one shot (T/F) ✗

- Syntax
  - → **Atomic** (single proposition)
    - eg:- 2+3=5, earth is flat.
  - → **Complex** (more than one P)
    - → $\sim$ Not / Negation
    - → $\vee$ Disjuction (or)
    - → $\wedge$ Conjunction (and)
    - → $\rightarrow$ implies (If then).
    - → $\leftrightarrow$ iff

Truth Tables

$\sim$

A	
0	→ 1
1	→ 0

$\wedge$

A	B	
0	0	→ 0
0	1	→ 0
1	0	→ 0
1	1	→ 1

$\vee$

A	B	
0	0	→ 0
0	1	→ 1
1	0	→ 1
1	1	→ 1

$\rightarrow$

A	B	
0	0	→ 1
0	1	→ 1
1	0	→ 0
1	1	→ 1

$\leftrightarrow$

A	B	
0	0	→ 1
0	1	→ 0
1	0	→ 0
1	1	→ 1

Predicate Logic

→ Use of variables (one or more)

eg:- ① x is equal to 5 : $P(x)$

② $x = y$: $P(x, y)$

③ x is married to y : $P(x, y)$

④ x is a integer : $P(x)$

Quantifiers

→ Universal "$\forall$" (all, Every)

→ Existential "$\exists$" (some, few, any)

* $P(x)$ is true for all $x \in A$

↓

$\forall x \in A, P(x)$

↑

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

$P(x)$: x is less than 5 ✓

If → Domain: $\{1, 2, 3, 10\}$

↳ $\exists x \in A \; P(x)$

$P(x)$ is true for some $x \in A$.

- First order Predicate logic

"$\forall$" "$\exists$"

All boys love cricket

$$\Downarrow$$

$\forall x : boys(x) \rightarrow love(x, cricket)$

Some boys hate books

$$\Downarrow$$

$\exists x : boys(x) \wedge hate(x, books)$

→ Negation of Quantifiers : ($\sim$)

$P(x)$: x is prime number

$$\sim \forall x \, P(x) \Rightarrow \exists x \sim P(x)$$

$$\sim \exists x \, P(x) \Rightarrow \forall x \sim P(x)$$

- Planning
  - → Reasoning side of Acting
  - → $(S) \longrightarrow (D)$
    - ↳ Actions involved
    - "Why"? ✓
  - → Its about deciding the tasks/actions to be performed by AI sytem.

→ **Plan**: set of action

↓ To be performed in a

sequence

& ↓ each

Action has its own

"preconditions" ← I/P

&

"effects" ← O/P.

Types

- → forward state space planning

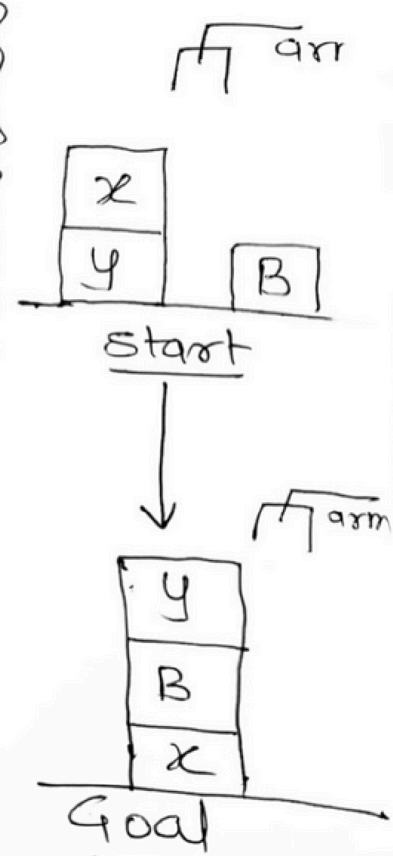
  Initial state $\xrightarrow{\text{Progress}}$ Goal state

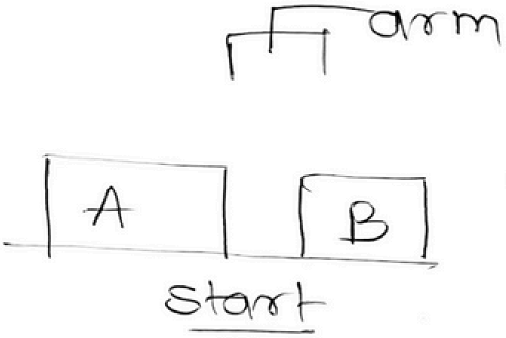
- → Backward state space planning

  Goal state (G) $\xrightarrow{\text{Regression}}$ Sub-goal (g)

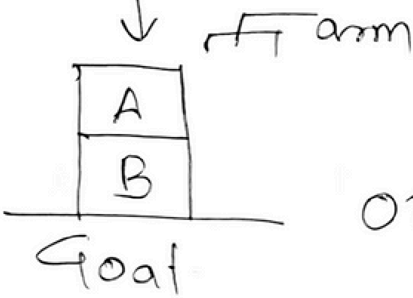
Now, we combine the features of FSSP & BSSF to get Goal stack planning.

	Preconditions	Effect
① Pick up(x) →	∘ arm empty ∘ On (x, table) ∘ clear (x)	∘ holding (x)
② Putdown(x) →	∘ holding (x)	∘ arm empty ∘ On(x, table ∘ clear (x)
③ Stack (x, y) →	∘ holding (x) ∘ Clear (y)	∘ On (x, y) ∘ clear (x) ∘ Arm empty
④ Unstack (x, y) →	∘ On(x, y) ∘ clear (x) ∘ arm empty	∘ holding (x) ∘ clear (y)





on(A, table)
on(B, table)



on(A, B)

On(A, B)

stack(A, B)
- holding(A)
- clear(B)

pickup(A)
- arm empty
- On(A, table)
- Clear(A)

① Pickup(A)
② Stack(A, B)

- Natural Language Processing (NLP)
  - → Understand, Analyze & manipulate "Natural Languages"

→ Components
- → NLU
- → NLG

① NLU
- → Lexical Ambiguity (word level)
- → Syntactical Ambiguity (sentence level)
- → Referential Ambiguity (Referring issue)

② NLG
- → Text planning
  (Retrieve Relevant content from KB)
- → Sentence planning
  (words, meaningfull phases, setting tone)
- → Text Realization
  (mapping sentence planning into sentence structure)

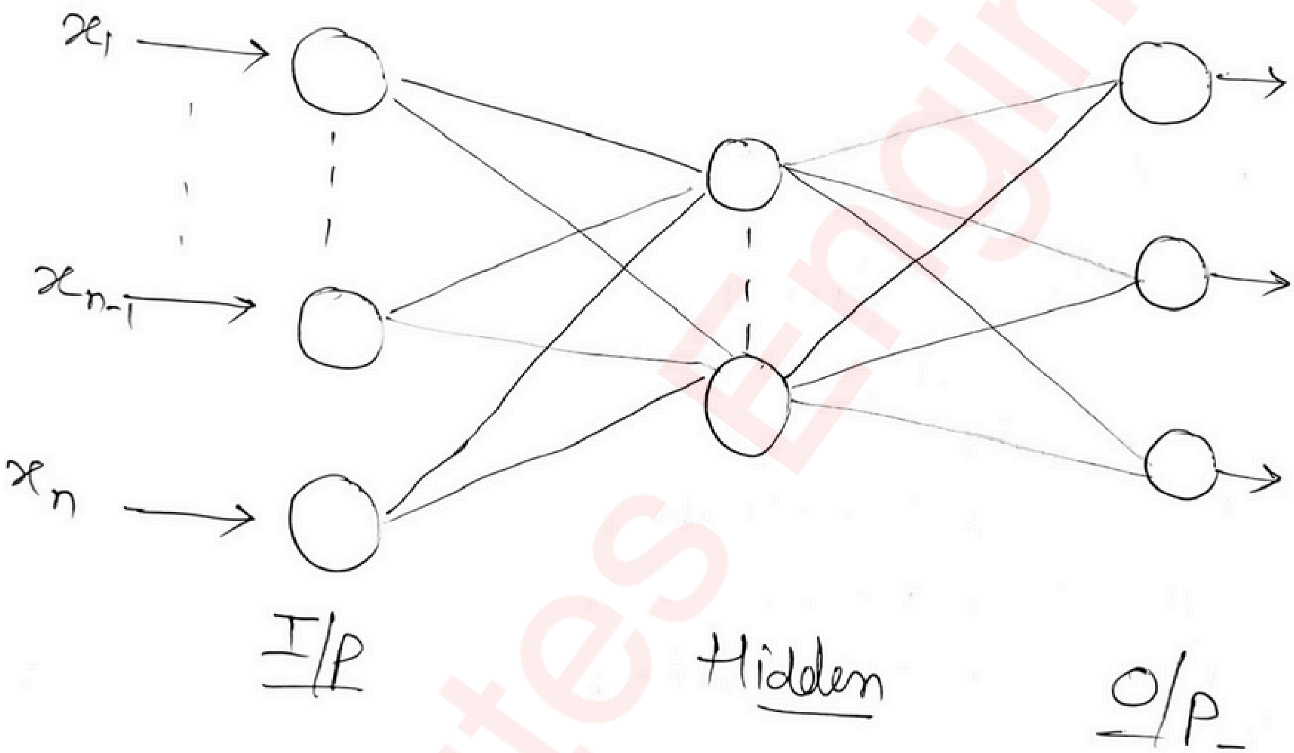
- Levels / phases of NLP
  - → Speech Analysis
  - → Lexical Analysis (lexicons/words)
  - → Syntactic Analysis (Relationship among words)
  - → Semantic Analysis (meaningfulness "hot ice-cream")
  - → Disclosure Analysis (Before & after sentences
  - → Pragmatic Analysis (context matters)

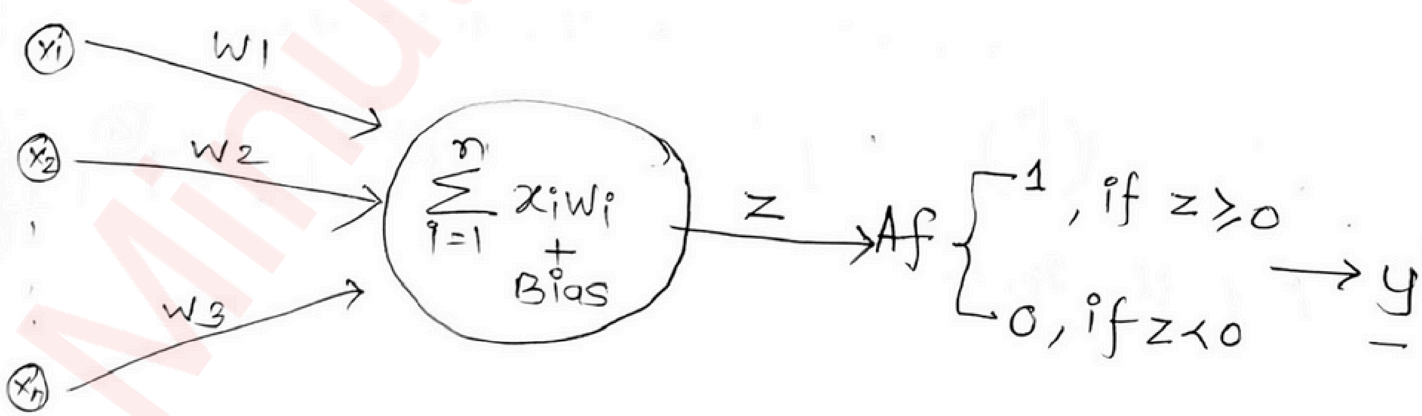
- Basic Terminologies
  - → Tokenization
  - → Stemming
  - → Lemmatization
  - → POStagging
  - → Stopwords
  - → Corpus
  - → vocabulary

Applications
- → Speech Recognition
- → Sentiment Analysis
- → Machine Translation
- → Text summarization
- → Text classification.

Neural Networks

- mimic human brain.
- Layers
  - Input
  - Hidden
  - Output





Genetic Algorithm

- → Adaptive heuristic search algo. inspired by "DTOE". (Genetics & natural selection)
- → To generate high quality soln for optimization problem.

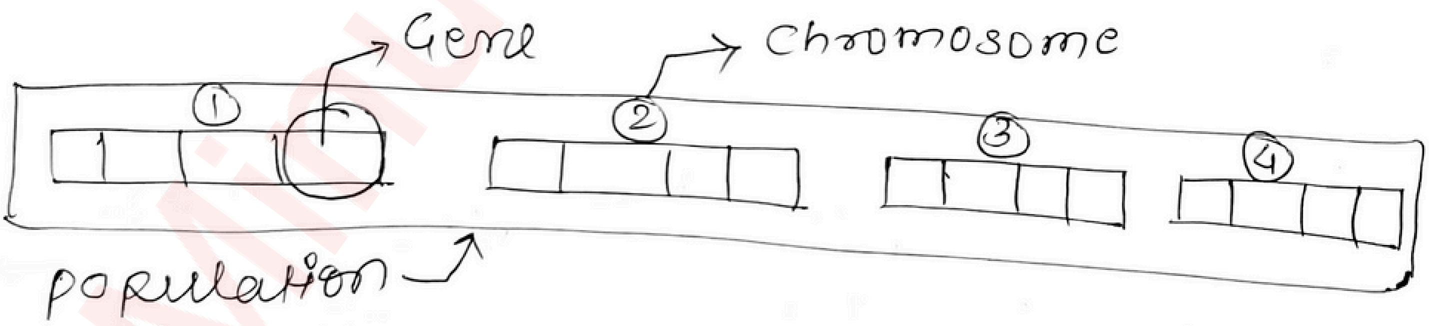
→ **Terms**

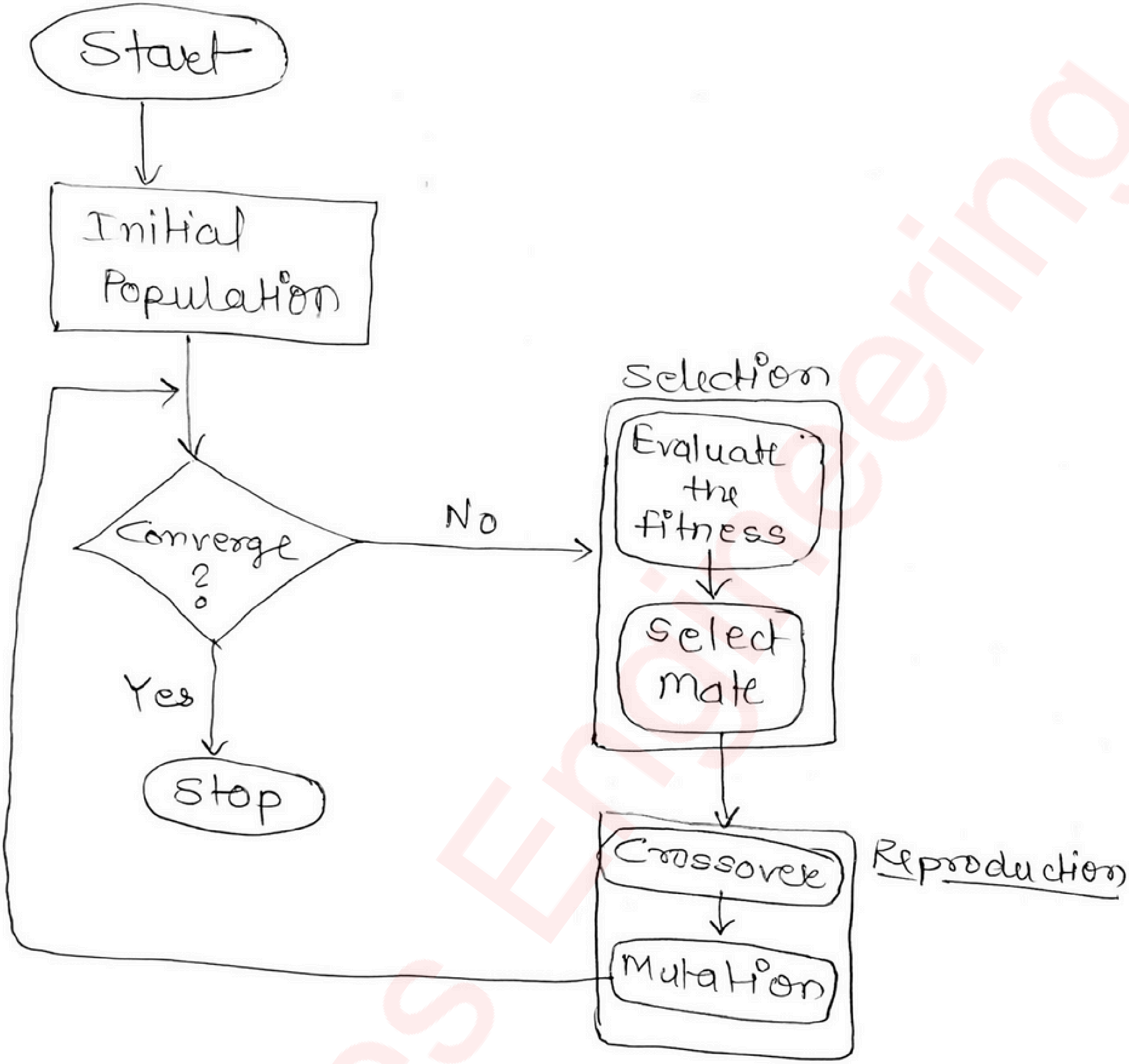
- → Population
- → Chromosomes
- → Gene
- → Allele
- → fitness function
- → Operator of GA
  - → selection
  - → crossover
  - → mutation

Phenotype

Decode ↑ ↓ Encode

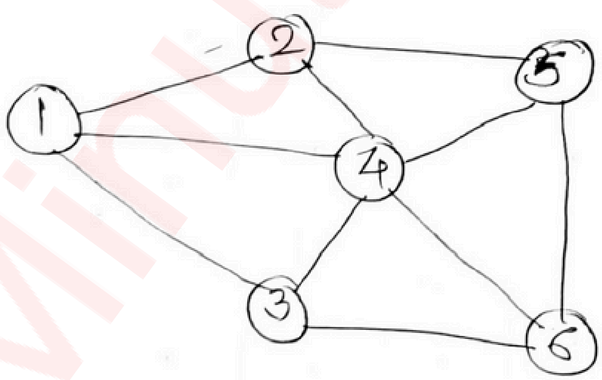
Genotype





fitness function :

eg:-



P1 : 1 2 5 6 4 3 2 1 → 18

P2 : 1 2 5 4 6 3 1 → 20

P3 : 1 2 4 5 6 3 1 → 12

P4 : 1 2 5 6 3 4 1 → 17

Crossover

P1 :	0	1	1	0	1	0
P2 :	1	1	0	1	0	0

single point

O1 : 0 1 0 1 0 0

O2 : 1 1 1 0 1 0

2 point

O1 : 0 1 0 1 1 0

O2 : 1 1 1 0 0 0

Multipoint

O1 : 1 1 0 1 1 0

O2 : 0 1 1 0 0 0

Mutation

Offspring :	0	1	1	0	0	1
MP (Probability) :	0	1	0	0	0	1
MO :	0	0	1	0	0	0

or Interchanging or Exchange ⇒ 0 1 1 0 0 1

0 0 1 0 1 1