Artificial Intelligence

20DCS103

Date
Page

* AI : The science and engineering of making intelligent machines. [machine + human intelligence]

** AI → Artificial : Man-made
↘ Intelligence : Power of thinking.

** Reasons of boost in AI :-

1) Softwares or devices can be made to solve real time problems.
2) Creation of virtual assistants. [Alexa, Siri, ...]
3) Robots development. [Helps in dangerous environment conditions]

** Goals of AI :-

1) Replication of human intelligence.
2) Solving problems which requires knowledge.
3) Building a machine that can do human intelligence task. [Playing chess, proving theorems, automated driving cars, ...]
4) Providing advice to the user.
5) Intelligent connection b/w perception and connection.

** Applications of AI :-

1) Gaming : Chess, Tic-Toe, Poker.
→ machine can think large number of moves,

2) NLP (Natural Language Processing) :
→ machine can understand human language.

3) Health care
→ Fast diagnosis.
→ Robotic surgery.

4) Finance. : Adaptive intelligence.
→ automatic chat box.
→ algorithm trading.

5) Data security :
→ AI helps in making data more secure.
→ AEG (Automatic Exploit Generation)
AI2 (Analyst in the loop) platforms are used to decide programming bug and digital assaults in a superior manner.

6) Expert systems :
→ Integration of software, machine and special information to provide reasoning and advice.

7) Computer vision :
→ Understanding the visuals automatically by machines.
→ Image processing.
→ Amazon go.

8) Speech recognition :
→ Extracting the meaning of the sentence.

→ 3. Noise removal ....

9) Robotics :-

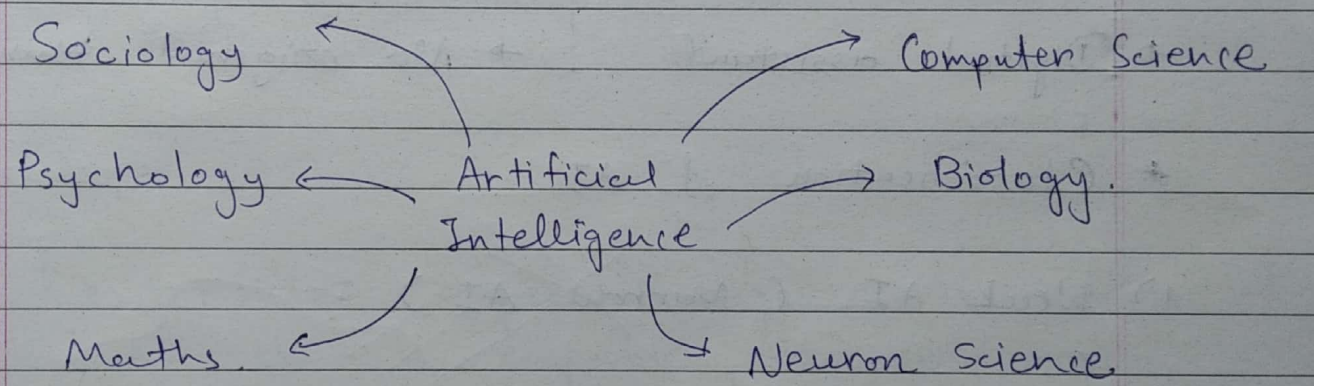
→ Ex:- Erica and Sofia : They can talk like humans.

10) E-commerce :-

→ 1. Automatic recommendation of products by analyzing their behaviour

→ 2. Service request handling.

* AI is comprised of ....



→ These all are the combination of,

Reasoning → a way to infer facts from existing data.

Learning → inducing a general rule from a set of observation.

Problem-solving → systematic search to reach to goal

Perception → the process of interpreting vision, sounds, smell and touch.

* Advantages	* Disadvantages
→ Accuracy ↑ Error ↓	→ cost ↑
→ Fast decision making.	→ Can't think beyond the limits.
→ Reliability is more.	→ No feeling & emotions.
→ Usefulness in risky areas.	→ more dependency on machines.
→ Digital assistant.	→ No original thinking.

* Classification of AI :-

1) Weak AI (Narrow AI) :-

↳ → Able to perform dedicated task with intelligence.
→ Can't perform beyond it's field or limitations.
→ Not concerned with how tasks are performed but it's concerned with the performance and efficiency.
→ Ex :- Flying machine.
Apple siri
Playing Chess

2) Strong AI (General AI) :-

↳ Study and design of machines that simulate human mind to perform intelligent tasks.
→ Can be built by borrowing ideas from psychology and neuroscience.
→ Forgetting things, understanding languages, Genetics, ...

3) Evolutionary AI :-

↳ Study and design of machines that simulate simple creatures and attempt to evolve.
→ Ex:- Ants, Bees, etc...

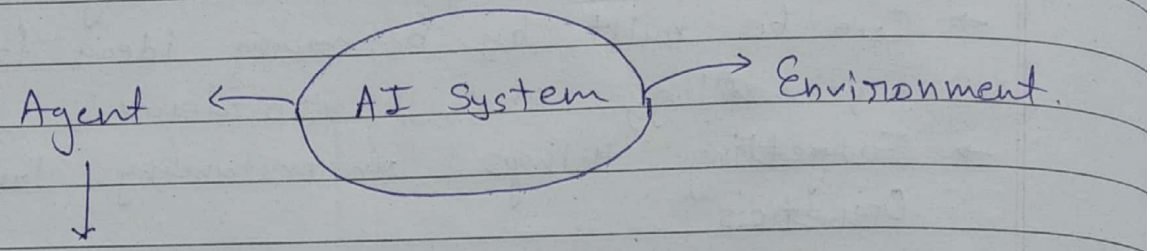
4) Super AI :-

↳ Artificial Super Intelligence is a form of AI that is capable of surpassing human intelligence by manifesting cognitive skills and developing thinking skills on its own.
→ machines > humans

* Agents :-

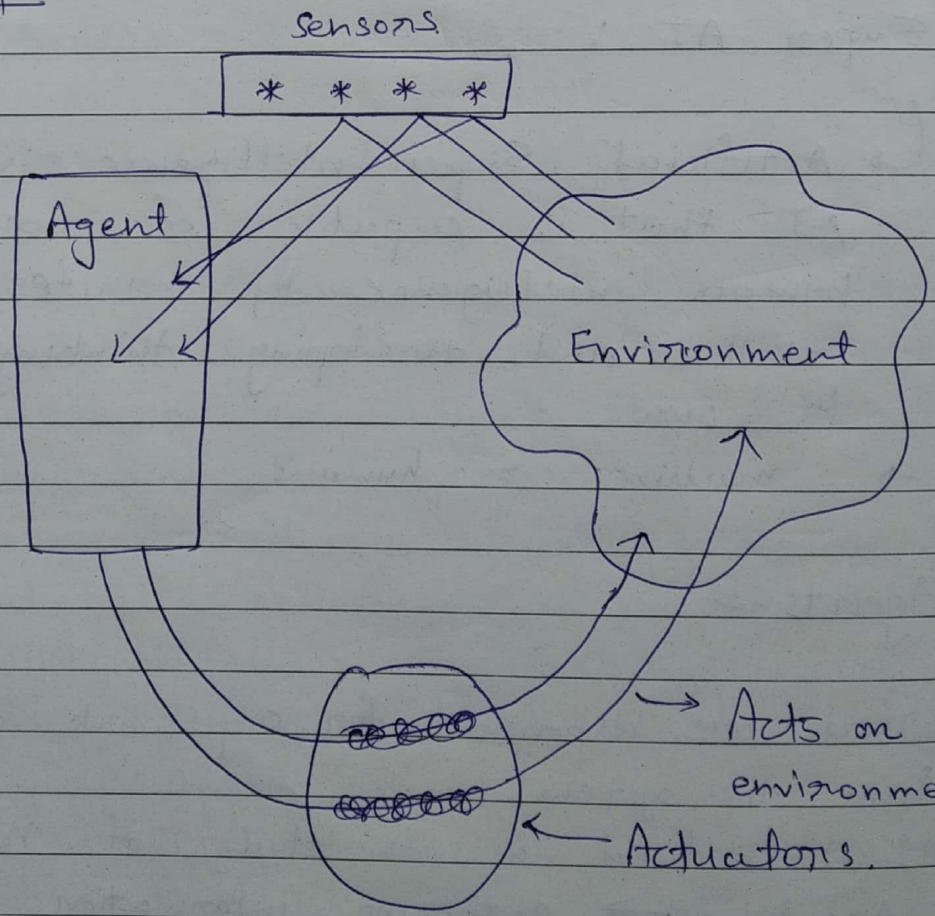
→ Which are responsible for any work output obtained from the system.
→ AI is defined as the study of rational agents.
→ Agent → gathering information
→ performing actions on it.

→ Rational agents → Person
→ Firm
→ Machine / Software } which can make decisions.



→ perceiving its environment through sensors.
→ Acting upon that environment through actuators.
↳ component of machines that converts energy into motion.

→ Example :-



→ Air conditioner is the best example.

→ Environment :- it's a part of the universe that surrounds the intelligent system.

* Tasks in AI :-

→ 1) Mundane tasks :-
→ Human learns mundane (Ordinary) tasks since their birth.
→ Can be learned by speaking, perception, using languages, etc...
→ Ex:- recognizing people, communicating through natural language, etc.

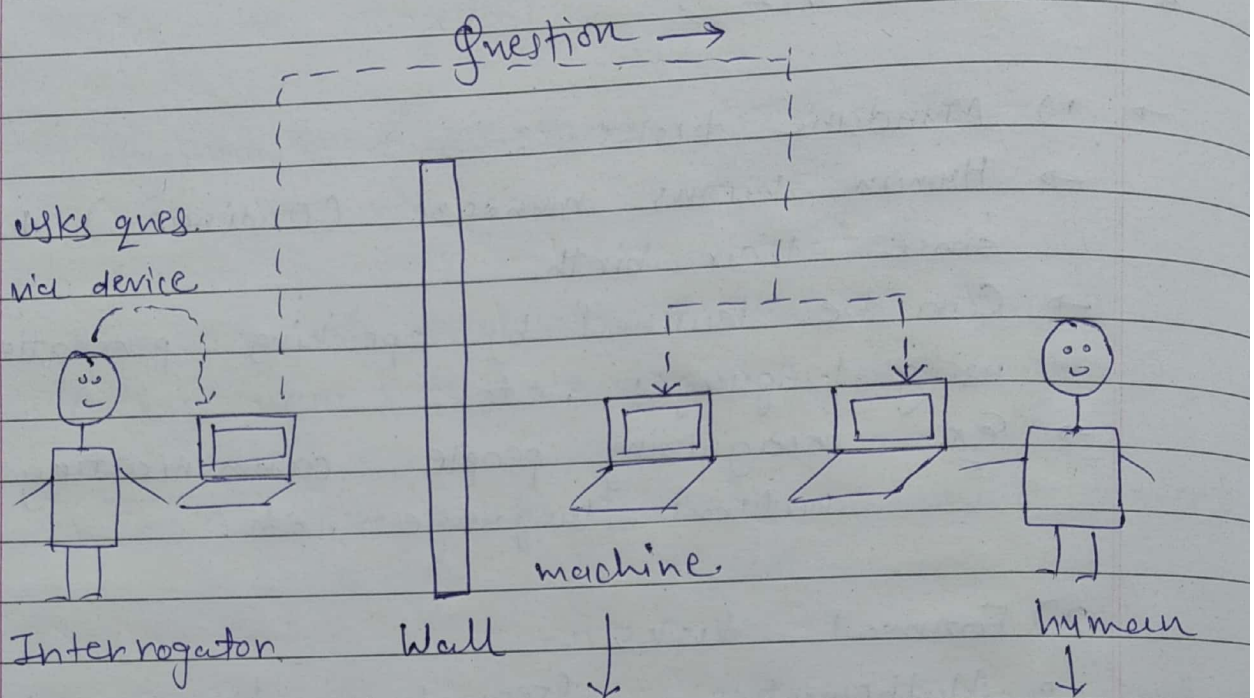
2) Formal tasks :-
→ Mathematics, Geometry, logic, Game-theory, etc..
→ verification and proving theorems.

3) Expert tasks :-
→ Engineering, manufacturing, monitoring, scientific, financial and medical fields.

* Turing test :- (discovered in 1950) by Alan Turing.
→ 'Turing test is used to determine whether or not machines can think like humans!'

→ Basic configuration :-
→ There will be a human interrogator or Examiner on one side of wall and other side a machine and human.
↳ knowledge base.

→ Machine is intelligent when human interrogator can NOT distinguish the responses given by machines and human.



if both are same or if interrogator can't distinguish b/w them, then machine is intelligent

* Chinese Room test :- (first published in 1980 by John Searle (Chinese Room Argument)

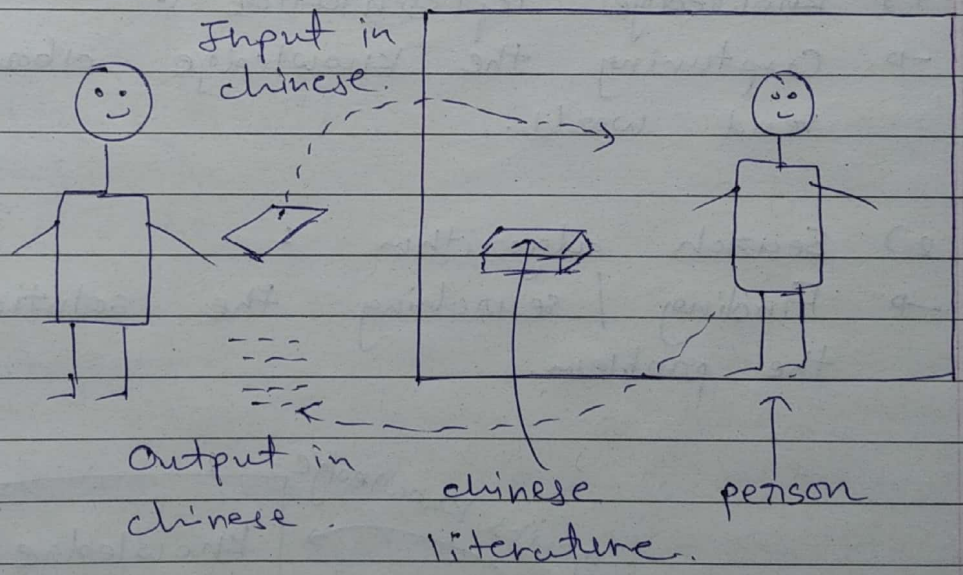
→ John Searle argued that 'Turing test could not be used to determine whether or not machine is considered as intelligent.'

→ A machine can pass turing test simply by manipulating symbols without having any understanding of those symbols. Therefore, machine can not be called intelligent.

→ A person / machine can be considered as intelligent, if and only if they do have the understanding of what they are doing.

→ Basic configuration :-

→ A person knowing english (not chinese) sits in a room with huge volume of chinese literature.



→ If a person giving the answer from the room is not considered as intelligent as person doesn't know the meaning of that chinese word / answer / question.

* Loebner contest : most human like conversation system.

* AI technique :-

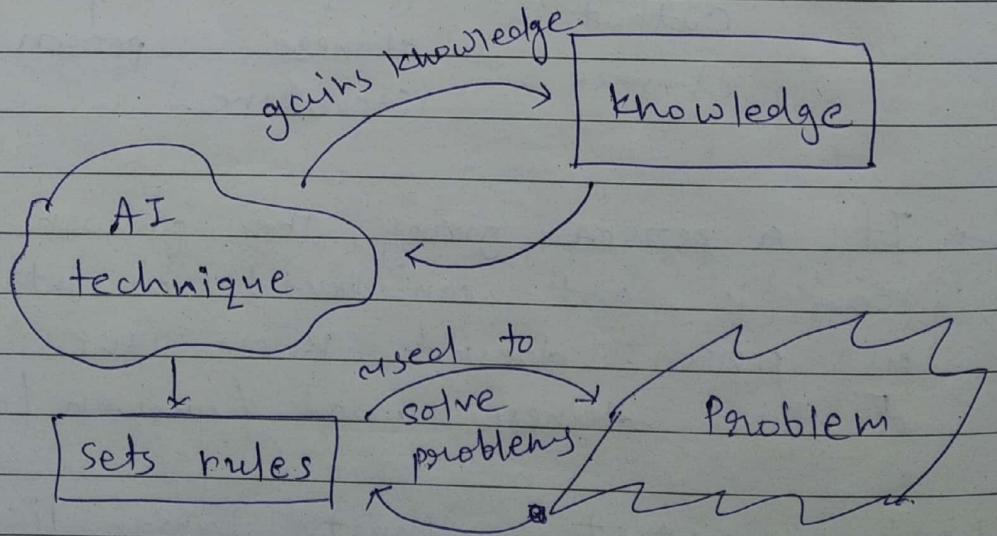
→ It is a method that exploits knowledge that should be represented in such a way that,

i) knowledge captures generalization.
ii) Understandable by people.
iii) Easily modified to correct.
iv) Can be used in many situations.
v) Can reduce its volume.

* Parts of AI technique :-

1) knowledge representation :
→ Capturing the knowledge about the real world.

2) Search algorithm :
→ Finding / searching the solution of the problem.



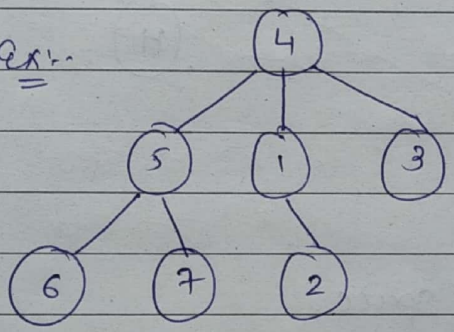
* Characteristics of knowledge :-

→ large in volume.
→ not well formatted
→ constantly changing.

* Breadth First Search (BFS) :-

→ Can start traversal from any sorting vertex.
→ When a vertex is visited, it should be explored completely.
→ Neighbouring vertices can be visited in any order.

→ Ex :-



BFS : 4, 5, 1, 3, 6, 7, 2.

→ Advantages :-

→ Definitely finds a solution is it exists.
→ finds minimal solution in least no. of steps.

→ Disadvantages :-

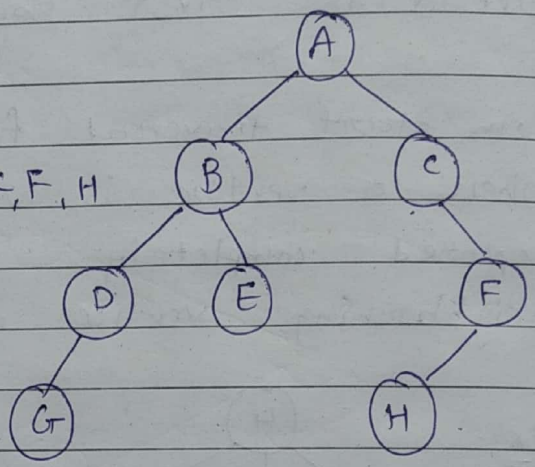
→ Consumes more memory.
→ Needs lot of time when sol^{n} is far from root node.

* Depth First Search (DFS) :-

→ Recursive algorithm.
→ Starts from root node and follows each path to its greatest depth node before visiting / moving to the next path.

→ Ex:-

DFS : A, B, D, G, E, C, F, H



→ Advantages :

→ consumes less memory space.
→ It will reach at the goal node in a less time than BFS if it traverses in a right path.

→ Disadvantages :

→ No guarantee of finding a solution.
→ may go in infinite loop.

* Heuristic search :-

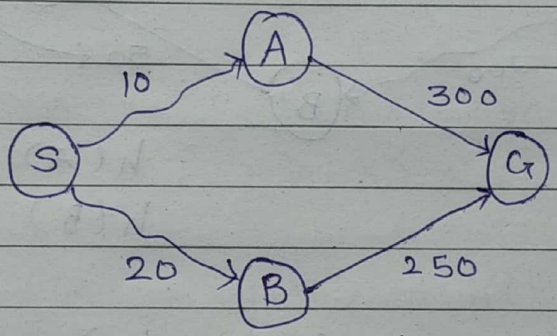
→ It tries to optimize a problem using heuristic function.

→ Heuristic function :-

It is a function $h(x)$ that gives an estimation on the cost of getting from node 'n' to the goal state.

→ Heuristic function helps in selecting optimal node for expansion.

→ Ex :-



$\therefore f(n) = g(n) + h(n)$

for A, $f(A) = g(A) + h(A)$
$= 10 + 300 = 310.$

for B, $f(B) = g(B) + h(B)$
$= 20 + 250 = 270$

∴ Root of node 'B' is the optimal soln to reach to node 'G'.

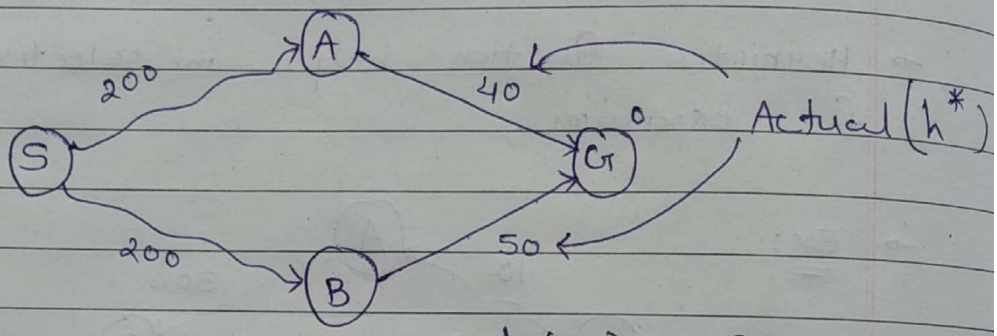
* Types of Heuristic search :-

1) Non - admissible : [Over-estimation]

→ $h(n) \geq h^*(n)$

↑ estimated ↑ actual

→ Ex :-



$g(A) = 200$ $h(A) = 80$

$g(B) = 200$ $h(B) = 70$

From $f(n) = g(n) + h(n)$

$f(A) = g(A) + h(A)$

$= 200 + 80 = 280$

& $f(B) = g(B) + h(B)$

$= 200 + 70 = 270$

} estimate cost

Now, $f(G) = g(G) + h(G)$

$= (200 + 50) + 0$ ← cost of reaching to G from G is always 0.

$= 250$

Actual cost

here we have considered (200 + 50) 250 of node 'B' because we find estimated cost lesser from node B.

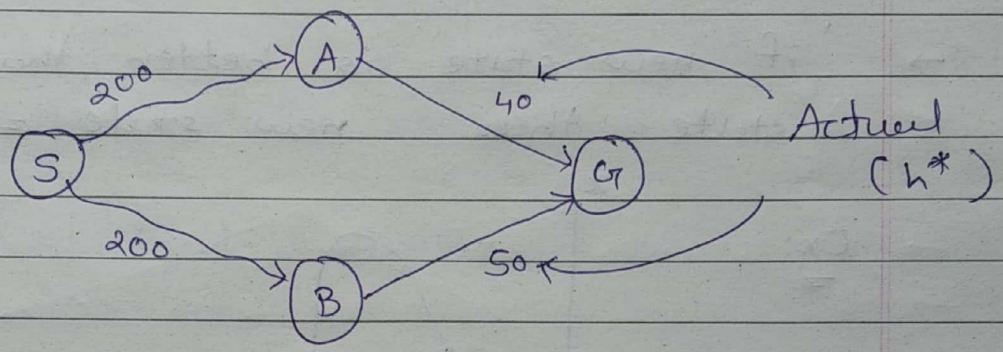
∴ $h(n) \geq h^*(n)$.

2) Admissible : [Under-estimation]

→ $h(n) \leq h^*(n)$

estimated actual

→ Ex :-



$g(A) = 200$ $h(A) = 30$

$g(B) = 200$ $h(B) = 20$.

From $f(n) = g(n) + h(n)$

$f(A) = g(A) + h(A)$

$= 200 + 30 = 230$

& $f(B) = g(B) + h(B)$

$= 200 + 20 = \boxed{220}$

} estimated cost

Now, $f(G) = g(G) + h(G)$

$= (200 + 50) + 0$

$= 250$

└ Actual cost

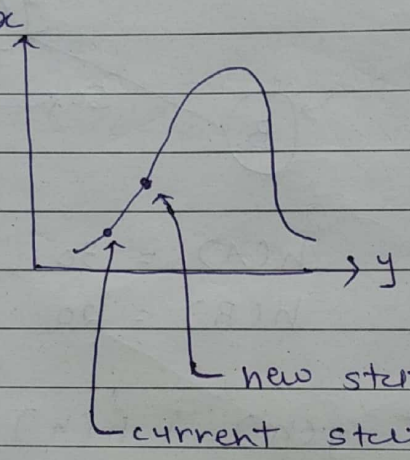
∴ $h(n) \leq h^*(n)$

* Hill Climbing Algorithm :-

1. Evaluate the initial stage.
2. Loop until a solution is found or there are no operators left.
   → Select and apply a new operator.
   → Evaluate the new state : if goal then quit

→ if new state is better than current state then new state = current state.

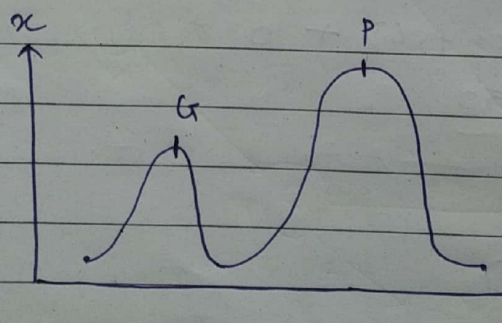
Ex:
=



here, new state is better than current state. ∴ new state is now current state.

* Disadvantages of Hill climbing algorithm:

1) Local maximum :-

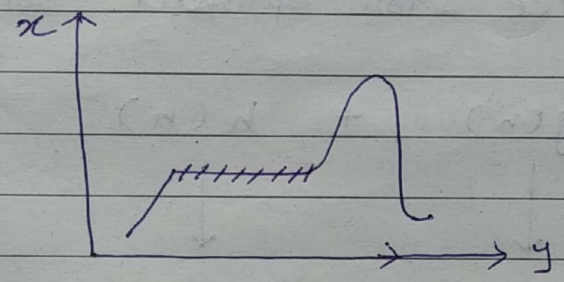


→ As hill climbing algorithm is local search algorithm, it will have the knowledge of local domain only. (Not global knowledge)

→ In this example, algorithm will stop at G because it is finding it's new state better than the current state up to point G.

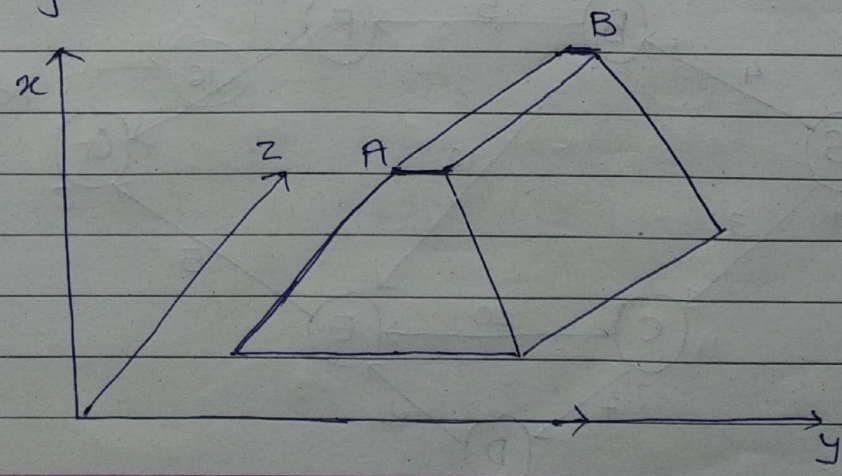
→ Hence, algorithm will not reach up to the next better state at point P.

2) Plateau / Flat maximum :-



→ Hill climbing algorithm will stop when it finds new state value exactly similar to the current state.

3) Ridge



→ In this case, hill climbing algorithm will NOT stop at point 'A' as it will find next state better but as of a structure, better state exists at point 'B'.

→ Hence, hill climbing algorithm will not give optimized result in case of ridge.

* A* algorithm : (Informed Searching)

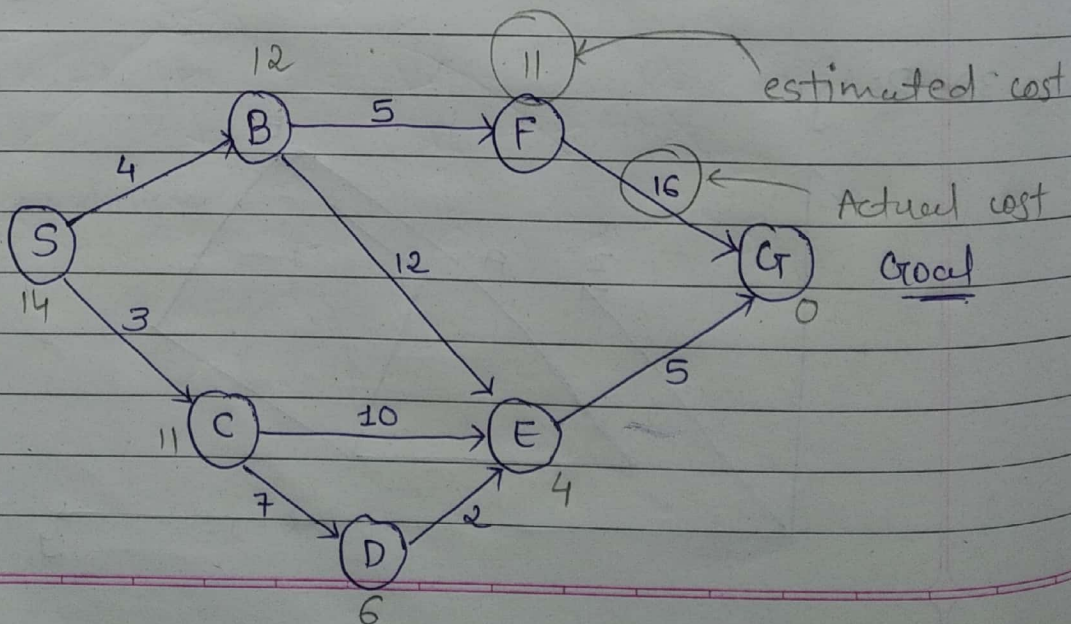
→ * stands for admissible characteristic.

→ A* gives guarantee for optimal sol^n.

$$f(n) = g(n) + h(n)$$

- $g(n)$: Actual cost from start node to n.
- $h(n)$: Estimated cost from n to goal node.

→ Ex:-



→ From, $f(n) = g(n) + h(n)$,

→ $f(S) = g(S) + h(S) = 0 + 14 = 14$

$f(S \to B) = g(S \to B) + h(S \to B)$

$= 4 + 12$

$= \boxed{16}$

$f(S \to C) = g(S \to C) + h(S \to C)$

$= 3 + 11$

$= \boxed{14}$

→ $f(SC \to E) = g(SC \to E) + h(SC \to E)$

$= (3 + 10) + (4)$

$= \boxed{17}$

$f(SC \to D) = g(SC \to D) + h(SC \to D)$

$= (3 + 7) + 6$

$= \boxed{16}$

Both are costing us need to explore Path

→ $f(SB \to F) = g(SB \to F) + h(SB \to F)$

$= (4 + 5) + 11$

$= 20$

$f(SB \to E) = g(SB \to E) + h(SB \to E)$

$= (4 + 12) + 4$

$= 20$

→ $SCD \to E = (3 + 7 + 2) + 4$

$= 16$

→ $SCDE \to G = (3 + 7 + 2 + 5) + 0$

$= 17$

→ Time complexity : $O(V + E) = O(b^d)$

where, b = branch factor

d = depth.

Space Complexity

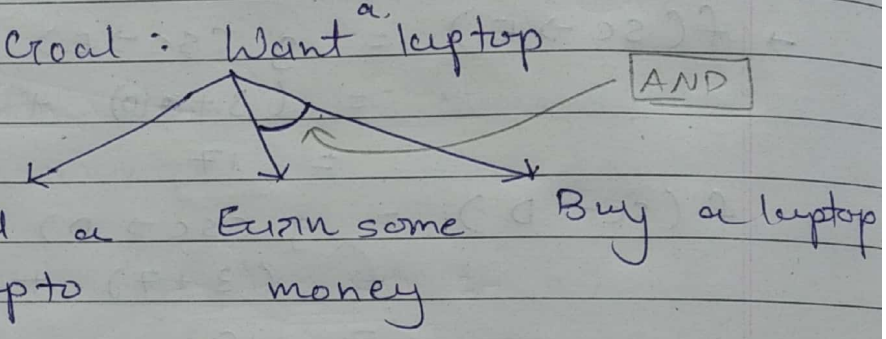
* AO* Algorithm : (Informed Searching)

→ AO* does NOT explore all the soln. paths once it got a solution.

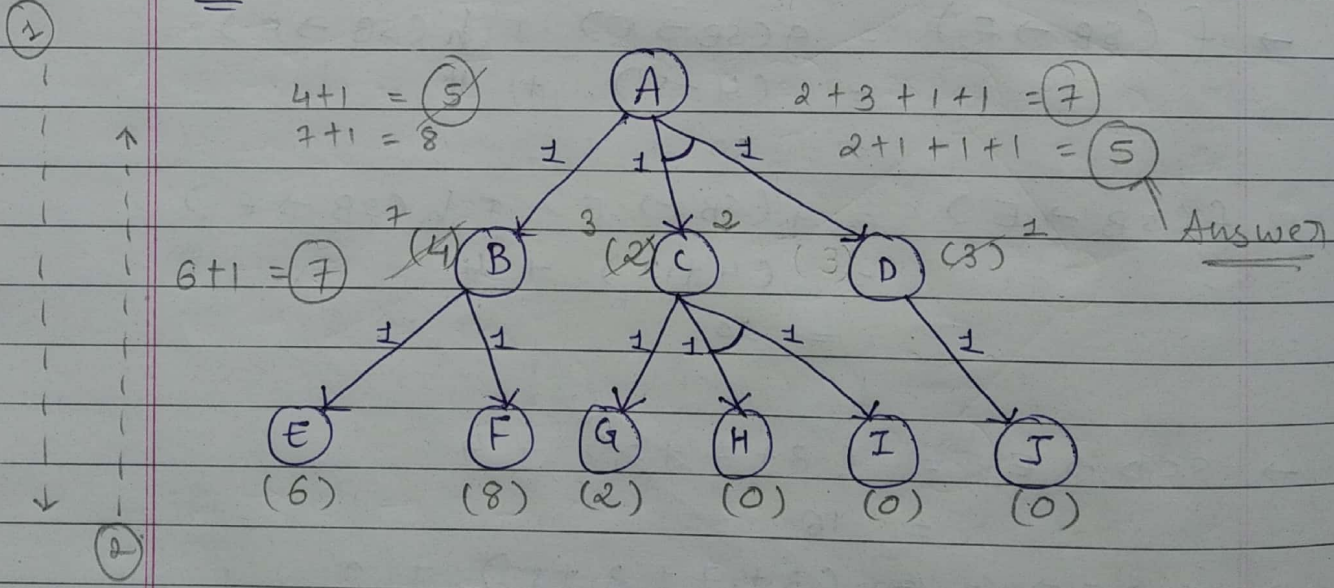
→ AO* (AND / OR) based algorithm having problem decomposition nature.

Breakdown into smaller pieces

→ Ex:



→ Ex:



* Best First Search (BFS) :-

→ Informed , Heuristic *

/* ** Algorithm :

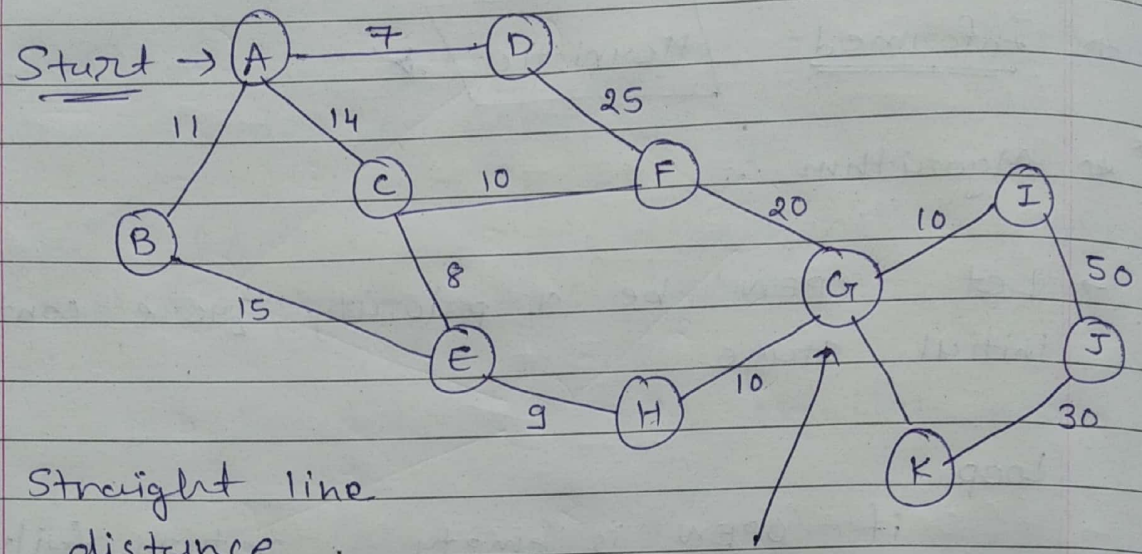
→ Let 'OPEN' be a priority queue containing initial stage.

Loop :
    if OPEN is empty , return failure.
    Node ← Remove first node from OPEN
        if node is a goal
            then return the path from initial to node
        else
            generate all the successors of node and put the newly generated node into OPEN according to their ~~fvalues~~ f values.
END Loop. */

→ 1) Begin with start node
2) Write heuristic value of each successors to the goal node.
3) Go with the one having least heuristic value.
4) Repeat from step (2).

→ Best First Search does not guarantee for optimal sol^{n}.

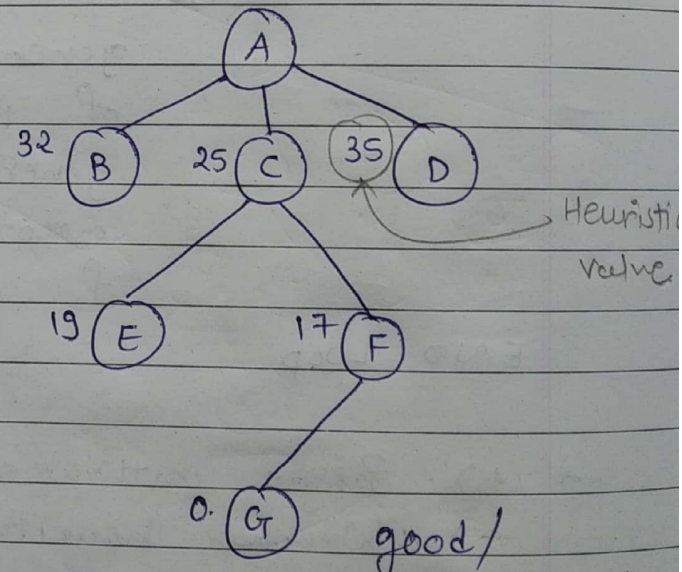
→ Ex :-



Straight line distance.
(Heuristic values)

A → G = 40
B → G = 32
C → G = 25
D → G = 35
E → G = 19
F → G = 17
G → G = 0
H → G = 0
I → G = 25
J → G = 32
K → G = 18

→ Answer :-



∴ A → C → F → G path gives ~~optimal~~ good/ sol^{n} by reaching to the goal node with minimum heuristic value.

→ Time complexity : $O(b^d)$

* Branch and Bound technique :-

→ It is a way to combine space saving of depth first search (DFS) with heuristic information.

→ Optimal sol^n is chosen among many sol^n path.

→ The idea of Branch and Bound search is to maintain lowest-cost path to the goal found so far and it's cost.

→ It is used with depth first search (DFS).

→ Two paths :→ Branch : Several choices are found.

:→ Bound : Setting bound on sol^n quality.

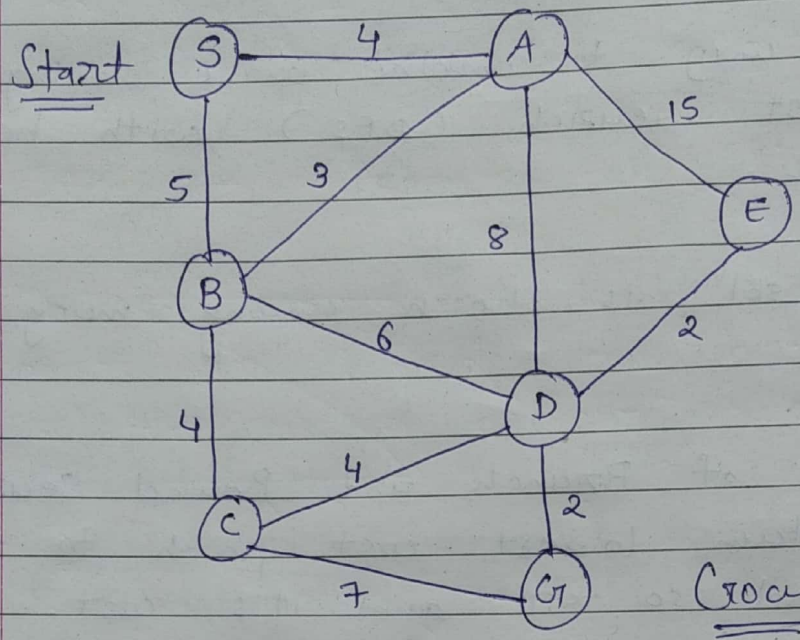
↳ Pruning : Trimming of branches where sol^n quality is poor.

→ It is based on the actual cost of reaching to the goal node.

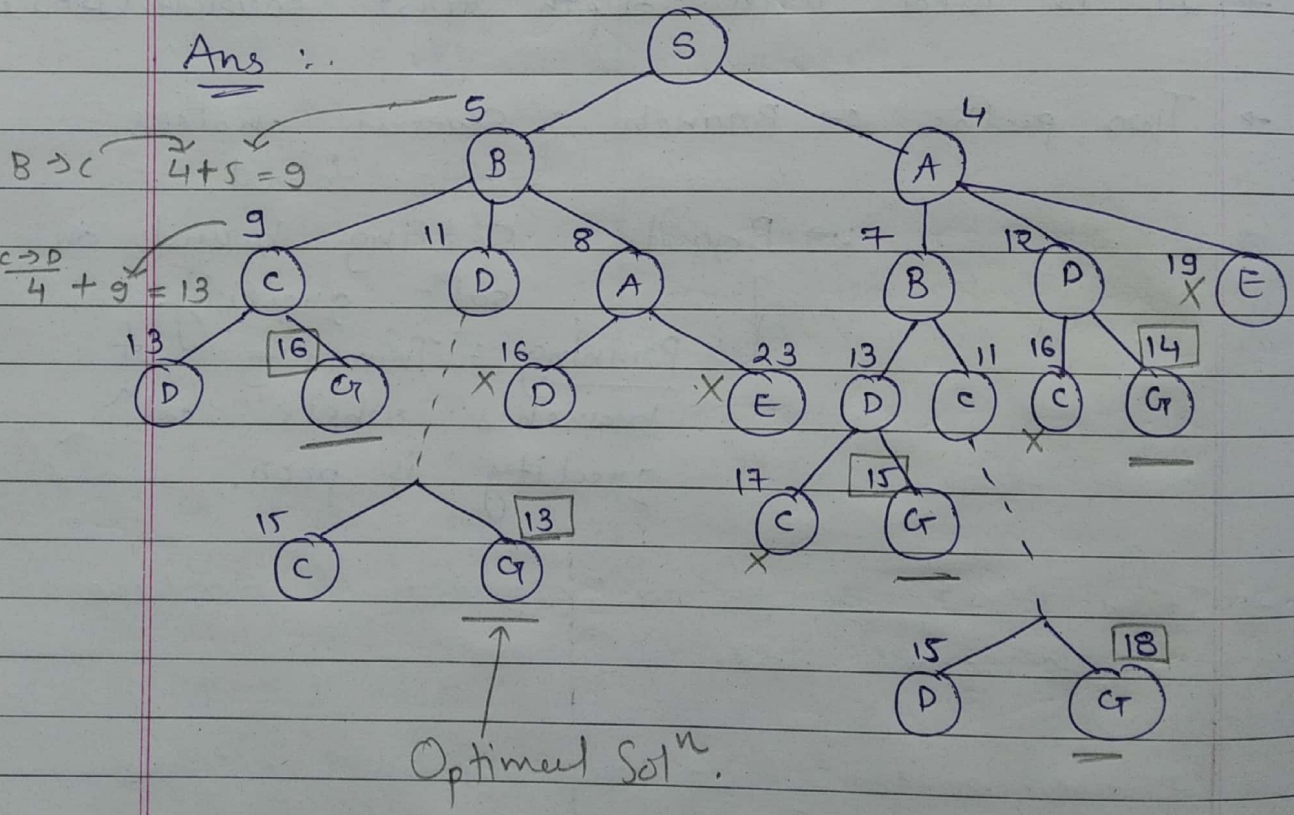
→ Always prefer least cost among all the possible cost of different paths.

→ If goal node is found with x cost then don't consider other parts consuming cost greater than or equal to x.

→ Ex :-



Ans :-

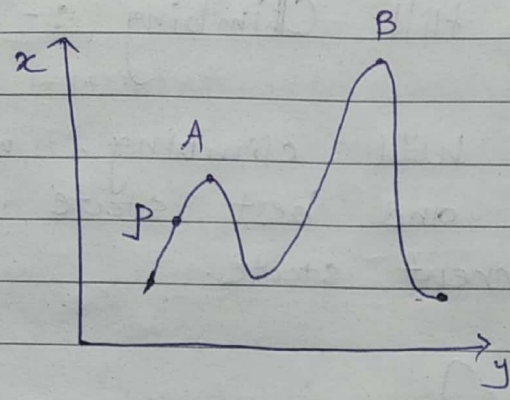


→ Optimal path : S → B → D → G

* Simulated Annealing :-

→ Allows downwards steps.

→ Checks all the neighbours.



→ When it reaches to point 'A', it will not find the next state better than the current state. Therefore it comes back to point p then new states will be analyzed with respect to p.

→ Here, it may reach to the 'B' where good solution is found.

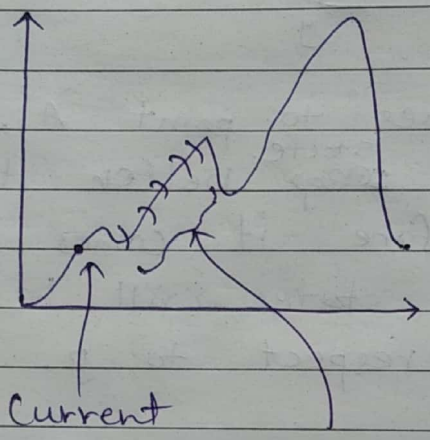
*	Advantages		Disadvantages
→	Easy to code for complex problems.	→	Slow process as it moves backwards.
→	Gives good sol^{n}.	→	Can't gurantee for optimal sol^{n}.
→	Statistically gurantees for optimal sol^{n}.	→	

→ 1) Annealing schedule is maintained.
2) Moves to worst state may be accepted.
3) Best state found so far is maintained.

* Steepest Ascent Hill Climbing :-

→ In steepest ascent hill climbing, multiple points are checked and best state is selected as a current state.

→ Ex:-



multiple points are checked, and the best one is selected as a new current state / next state.

→ All moves are considered and best one is selected as a new current state / next state.
→ It examines all neighbouring nodes and selects node closest to soln as new current node / next state.

* Constraint Satisfaction :-

→ It is a search procedure that operates in a space of constraint sets.

→ Constraint satisfaction problems in AI have goal of ~~satisfied~~ discovering problem state that satisfies a given set of constraints.

→ i) Constraints are discovered and propogated throughout the system.
ii) If still there is no sol^n, search begins.
↳ A guess is made about something and added as new constraint.

→ CSP consists of three components : V, D, C.

→ V is a set of variables $\{V_1, V_2, \ldots, V_n\}$.
→ D is a set of domains $\{D_1, D_2, \ldots, D_n\}$.
→ C is a set of constraints that specify allowable combination of values. $\{C_1, C_2, \ldots, C_n\}$

C_i = (Scope, Relation)

↓

Scope is set of variables that participate in constraint.

↓

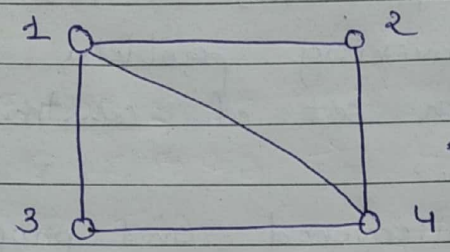
Relation defines the values that variable can take.

→ V_1 V_2 $C_1 = [(V_1, V_2), (V_1 \neq V_2)]$

domain A B $C_1 = ((V_1, V_2), (A, B))$

(1,2) (2,4) $C_1 = [(V_1, V_2), (1,2)(1,4)(2,4)]$

** Ex:- CSP problem :. (using intelligent Backtracking)



← Constraint graph

$V = \{1, 2, 3, 4\}$

$D = \{$ Red, Green, Blue $\}$

$C = \{1 \neq 2, 1 \neq 3, 1 \neq 4, 2 \neq 4, 3 \neq 4\}$

→ Neighbouring vertices should not have same colour.

Ans:-

	1	2	3	4
Initial domain	RGB	RGB	RGB	RGB
1 = R	R	GB	GB	GB
2 = G	R	G	G B	B
3 = B	R	G	B	___
3 = G	R	G	G	B

Conflict occurs as there is no legal color for 4th node according to constraint.

→ While taking blue color of 3rd node, we are unable to assign any color to the 4th node according to the constraints.

∴ Conflict occurs.

→ Therefore, we need to perfor intelligent back tracking which mean go back to that point which is responsible for the conflict.

→ In this case, assigning blue colour to the 3rd node is the reason for conflict.
∴ We need to check whether 3rd node can have any other color which satisfies the constraints or not.
If found, then assign that color to the 3rd node and follow the procedure for all the nodes.

**→ CSP : Example

→ Constraints : → No two letters have same value.
→ Sum of digits must be as shown in the problem.
→ There should be only one carry forward.

```
  S E N D
+ M O R E
---------
M O N E Y
```

Soln :→

C_4	C_3	C_2	C_1
1	0	1	1
	9	5	6	7
+	1	0	8	5
1	0	6	5	2

C_4	C_3	C_2	C_1
	S	E	N	D
+	M	O	R	E
M	O	N	E	Y

(1) $M = 1 \rightarrow S + M \geq 10 \quad (\because C_4 = 1)$

$\Rightarrow S \geq 10 - M$

$\Rightarrow S \geq 10 - 1$

$\Rightarrow S \geq 9$

$\therefore \boxed{M = 1} \Rightarrow \boxed{S = 9}, \quad C_4 = 1$

$O = 0 \quad C_3 = 0$

S	9
E	5
N	6
D	7
M	1
O	0
R	8
Y	2

(2) $E + O \leq 10 \quad (\because C_3 = 0)$

$E \leq 10 - 0$

$E \leq 10 \quad , C_2 = 1$

Assume, $E = 5 \quad \therefore E + O + C_2 = N$

$\therefore 5 + 0 + 1 = N$

$\therefore 6 = N$

$\therefore \boxed{E = 5} \Rightarrow \boxed{N = 6}, \quad C_2 = 1$

(3) $C_1 = 01 \quad , C_1 + N + R = E$

$\therefore N + R + C_1 = E$

$\therefore 6 + R + 1 = 15 \quad (\because C_2 = 1)$

$\therefore \boxed{R = 8}$

(4) $D + E = Y \qquad C_1 = 1$

$\therefore D + E \geq 10$

$\therefore D + 5 \geq 10$

$\therefore D \geq 5$

Now, $7 + 5 = Y \qquad$ Assume, $\boxed{D = 7}$

$\therefore \boxed{Y = 2} \quad (\because$ Carry $C_1 = 1)$

* 8 - puzzle problem :-

** Without heuristic : Blind search (Uninformed)

→ Breadth First Search
→ $O(b^d)$, b = branch , d = depth.
→ T. C. can go up to $O(3^{20})$ in worst case.
→ 4 moves are possible : Up, Down, Left, Right.
→ In this case, each node is explored and compared with the goal. If it's a goal then stop the further process otherwise continue finding other possiblities.

** With heuristic : (Informed)

→ Find the heuristic value for each node.
[heuristic value = no. of misplaced numbers]
→ Go for the node / Explore the node which has less heuristic value.
→ No gurantee for optimal sol^n. However, it can give good sol^n.

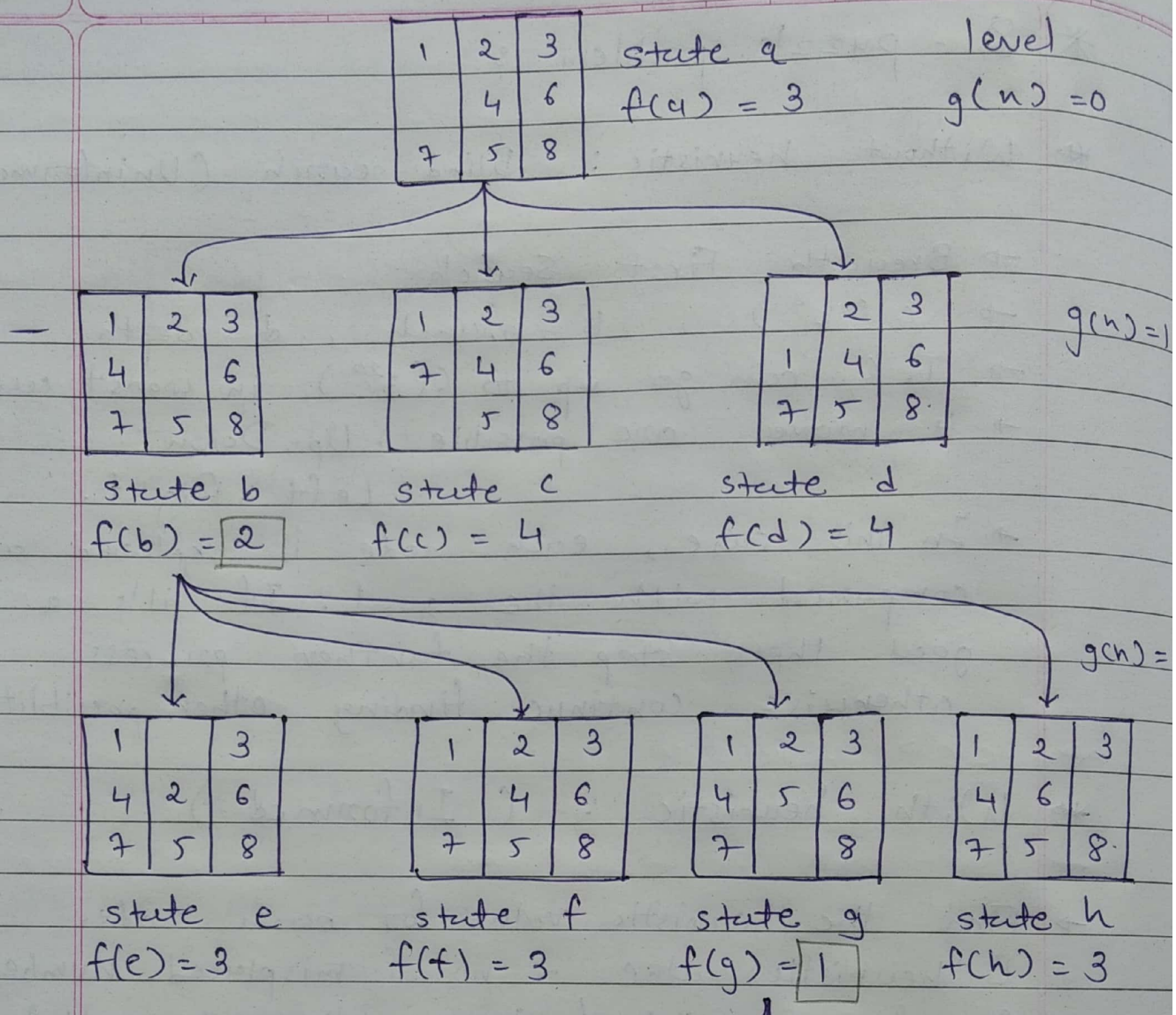
→ Ex :-

1	2	3
	4	6
7	5	8

Start

1	2	3
4	5	6
7	8	

Goal



* Knowledge Representation and Reasoning :-

→ Knowledge is a collection of specilized facts procedures and judgement rules.

→ Knowledge acquisition is the process of extracting, structuring and organizing knowledge from one or more sources.

* Propositional Calculus :-

→ It is a system that deals with the method used for manipulation of the symbols according to some rules.

* Alphabet set :-

→ i) Set of variables or Propositional Symbols
↳ P, Q, R
ii) Logical constants : True (T), False (F)
iii) Two parentheses : "(" and ")"
iv) Set of logical operators

↳

	Word	Symbol	Example. (X, Y)	
i)	Not	¬	¬X	: Not x
ii)	and	∧	X ∧ Y	: x and y
iii)	or	∨	X ∨ Y	: x or y
iv)	implies	→	X → Y	: if x then y
v)	if and only if.	↔	X ↔ Y	: if and only if x then y.

→ Examples :-

X : It is Hot.
Y : It is Humid.
Z : It is Raining.

1) If it is humid then it is hot.
→ $Y \rightarrow X$

2) If it is hot and humid then it is not raining.
→ ~~$\neg X \Rightarrow \neg Y$~~ $(X \wedge Y) \rightarrow \neg Z$

* Predicate logic :

→ Proposition : They are declarative statements which are either true or false but not both.

→ Quantifiers :
↳ They constrain the meaning of sentence counting a variable.
↳ Quantifiers are followed by a variable and a sentence.

→ Two quantifiers :-

1) Universal quantifier : $\forall x$
↳ for all x such that.
2) Existential quantifier : $\exists x$.
↳ for some x such that / there exists.

→ Examples :-

1) All boys like football.

→ $\forall x$: boys $(x) \rightarrow$ like $(x$, football$)$

2) Some boys like football.

→ $\exists x$: boys $(x) \wedge$ like $(x$, football$)$.

→ In universal quantifier, implication $(\rightarrow)$ is used as it is clear that all the x will like / likes football whereas in existential quantifier, and $(\wedge)$ is used as some of the x may or may not like football.

* Monotonic and non-monotonic reasoning :-

** Monotonic Reasoning :-

→ Once the conclusion is taken then it will remain same even if we add some other information to our existing information in our knowledge base.

→ In short, decisions are not affected by new facts added.

* Non-monotonic Reasoning :-

→ The conclusion changes by adding some

information to our existing information in our knowledge base.

** Fuzzy Logic :-

→ Fuzzy set theory : (Lotfi A. Zadeh)
  ↳ It is a mathematic tool for dealing with the concepts used in natural language.

→ Fuzzy logic :
  ↳ It is a multivalued logic that allows intermediate logic values to be defined between conventional equations.
  ↳ Mathematical tool to represent uncertainity and vagueness.

→ Characteristics :-

1) Exact reasoning is viewed as limited case of approximate reasoning.
2) Everything is a matter of degree.
3) Knowledge is interpreted as a connection of variables.
4) Any logical system can be fuzzified.

** Membership function in Fuzzy system :-

→ Membership function is used to represent degree of truth in fuzzy logic.

→ It characterizes fuzziness (all the info^n in fuzzy set.).
→ It is represented by graphical form.
→ It specifies the degree to which i/p belongs to set.
→ Degree of membership : o/p always lies b/w 0 to 1.

* Mathematical notations :-

For fuzzy set A,

$$A = \{ (x, \mu A(x)) \mid x \in X \}$$

$\mu A(x)$ → membership function
X → Universe of discourse, membership space.

* Set of $x \in X$, such that,

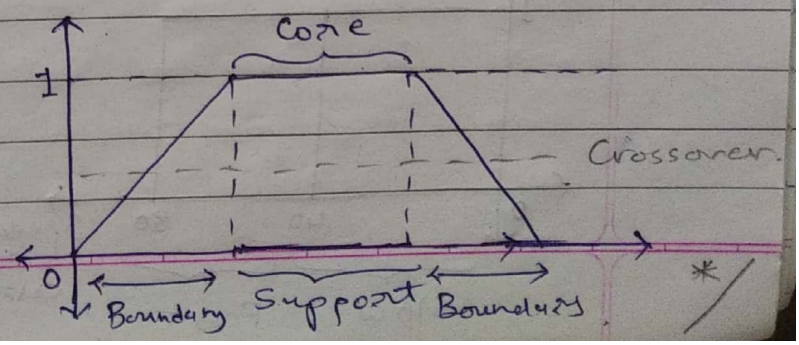
$\mu_A(x) = 1$ → Core.

$\mu_A(x) > 0$ → Support

$\mu_A(x) = 0.5$ → Cross over

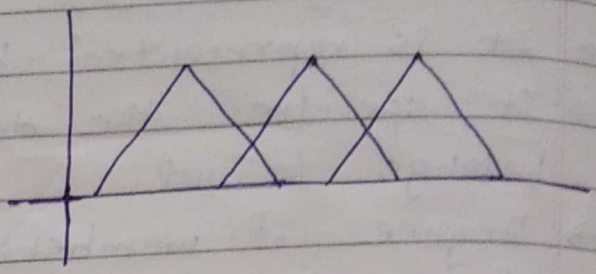
$0 < \mu_A(x) < 1$ → Boundary.

Graphical representation.

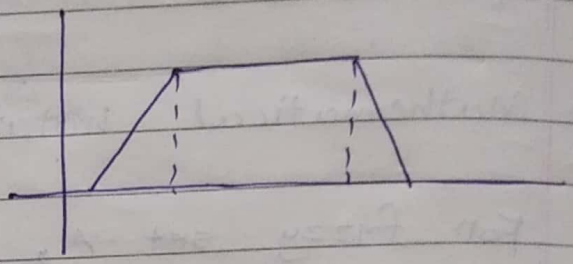


* Types of membership function :-

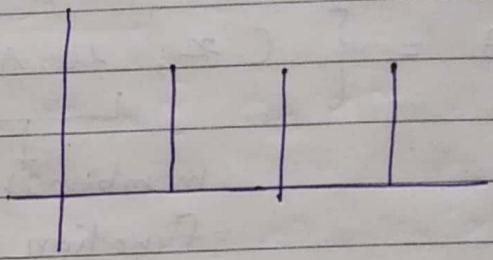
i) Triangular :



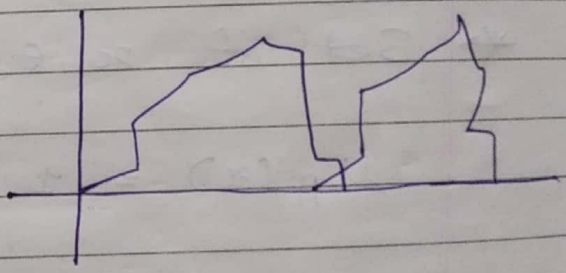
ii) Trapezoidal :



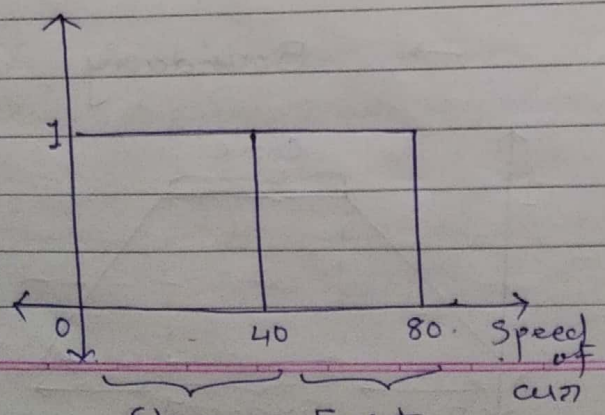
iii) Singleton :

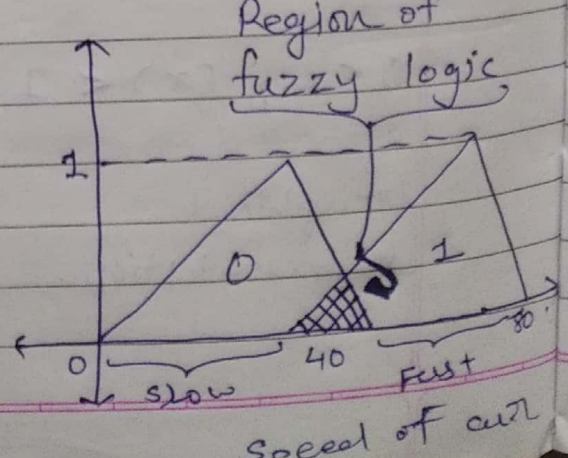


iv) Piece wise linear :



* Example :- Car speed > 40 => Fast
Car speed ≤ 40 => Slow





* Check the degree of fustness :-

$$(x, u) = \begin{cases} 0, & \text{if speed}(x) \leq 40 \\ \frac{\text{speed}(x) - 40}{10}, & \text{if } 40 < \text{speed}(x) \leq 50. \\ 1, & \text{if speed}(x) \geq 50 \end{cases}$$

→ $x = 30 \Rightarrow (x, u) = (30, 0)$

($\because$ 30 is less than 40).

→ $x = 55 \Rightarrow (x, u) = (55, 1)$

($\because$ 55 is greater than 50).

→ $x = 42$ (42 is b/w 40 and 50)

$\therefore u = \frac{\text{speed}(x) - 40}{10} = \frac{42 - 40}{10} = 0.2.$

$\therefore (42, 0.2)$ degree of fustness

→ $x = 41 \quad \therefore u = \frac{\text{speed}(x) - 40}{10}$

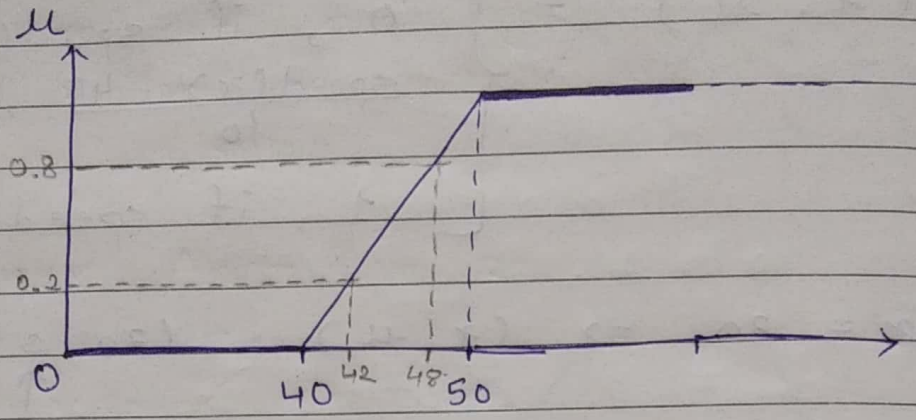
$= \frac{41 - 40}{10} = 0.1.$

$\therefore (41, 0.1)$ degree of fustness

→ $x = 48 \quad \therefore u = \frac{48 - 40}{10} = \frac{8}{10} = 0.8.$

degree of fustness

→ Graphical representation of this example :-



* Operations in fuzzy logic :- $x \in U$

OR 1) Union : $\mu_{A \cup B} = \text{Max}\{ \mu_A(x), \mu_B(x) \}$

AND 2) Intersection : $\mu_{A \cap B} = \text{Min}\{ \mu_A(x), \mu_B(x) \}$.

NOT 3) Complement : $\mu_{\bar{A}}(x) = [1 - \mu_A(x)]$

XOR 4) Bold Union : $\mu_{A \oplus B} = \underline{\text{Min}}[1, \mu_A(x) + \mu_B(x)]$

XNOR 5) Bold Intersection : $\mu_{A \cdot B} = \underline{\text{Max}}[0, \mu_A(x) + \mu_B(x) - 1]$

6) Equality : $A = B$ if $\mu_A(x) = \mu_B(x)$, $\forall x \in S$
for all x

→ If membership values (μ) are not given then membership function will be given and we have to find membership values from the function.

** Example :- $U = \{5, 10, 20, 25, 30, 40\}$

$A = \{(10, 0.2), (20, 0.4), (25, 0.7), (5, 0.2), (30, 0.9), (40, 1)\}$

→ ~~$\mu_{A \cup B} = \{$~~

$B = \{(10, 0.4), (20, 0.1), (25, 0.9), (30, 0.2), (40, 0.6)\}$

→ $\mu_{A \cup B} = \{(10, 0.4), (20, 0.4), (25, 0.9), (5, 0.2), (30, 0.9), (40, 1)\}$

→ $\mu_{A \cap B} = \{(10, 0.2), (20, 0.1), (25, 0.7), (30, 0.2), (40, 0.6)\}$

→ $\mu_{\bar{A}} = \{(10, 0.8), (20, 0.6), (25, 0.3), (5, 0.8), (30, 0.1), (40, 0)\}$

→ $\mu_{\bar{B}} = \{(10, 0.6), (20, 0.9), (25, 0.1), (30, 0.8), (40, 0.4)\}$

→ $\mu_{A \oplus B} = \{(10, 0.6), (20, 0.5), (25, 1), (5, 0.2), (30, 1), (40, 1)\}$

→ $\mu_{A \cdot B} = \{(10, 0), (20, 0), (25, 0.6), (5, 0), (30, 0.1), (40, 0.6)\}$

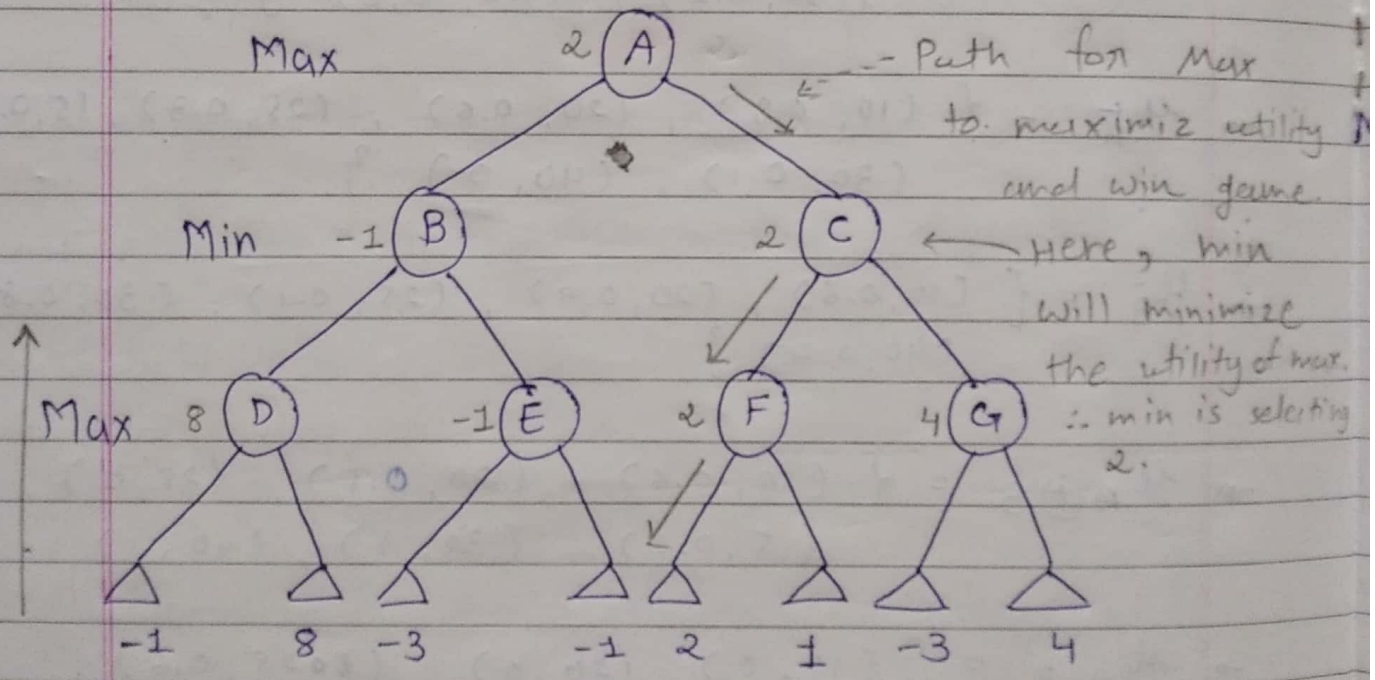
→ It is not equal.

* Introduction to Game Playing :-

→ Based on rules and logic.
→ Utility : kind of reward earned based on moves

* Minimax Algorithm :-

→ Backtracking algorithm.
→ Best move strategy used.
* → Max will try to maximize its utility. (Best move)
  → Min will try to minimize utility. (Worst move for max)



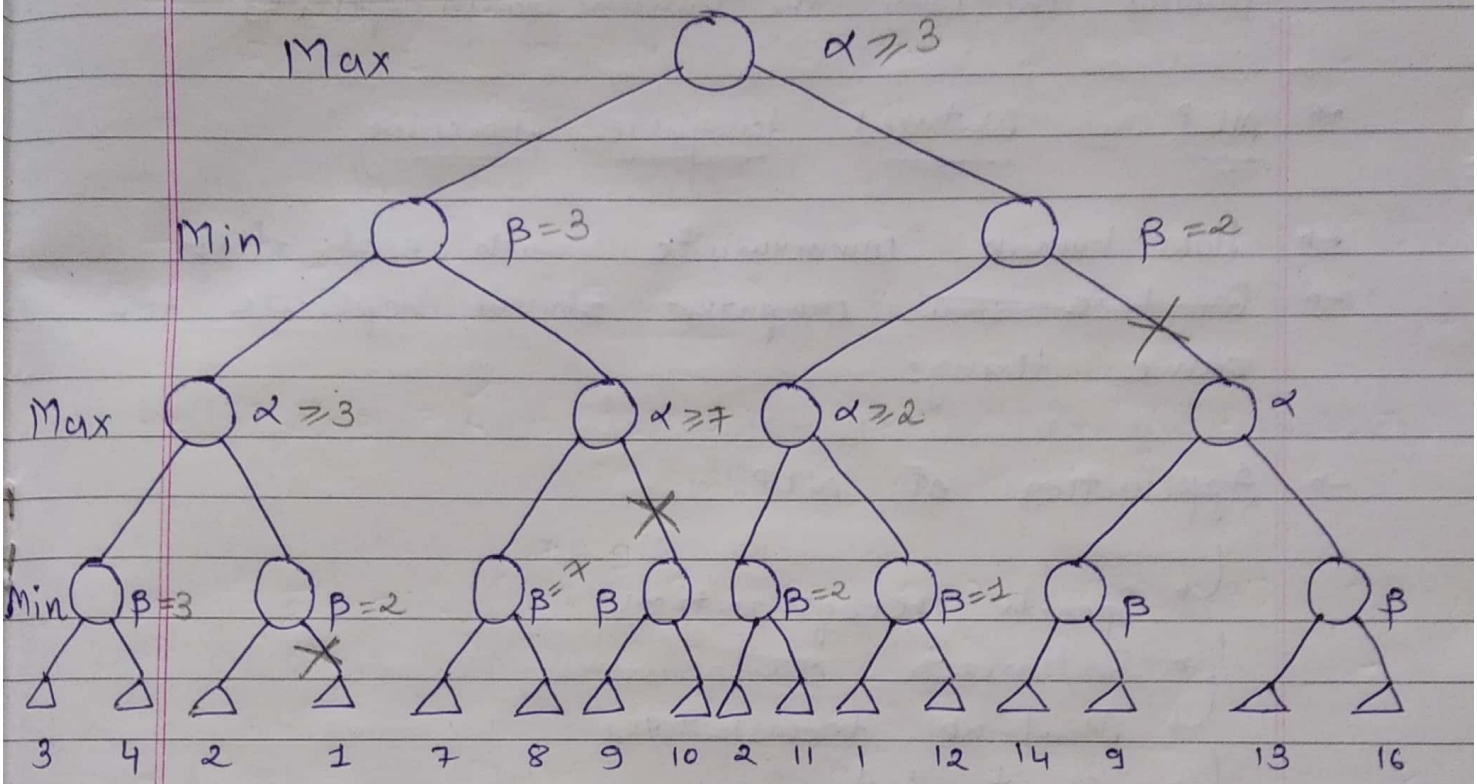
- Path for Max to maximize utility and win game.
← Here, min will minimize the utility of max. ∴ min is selecting 2.

→ For player Max : +ve ⇒ gain in Utility
and -ve ⇒ loss in Utility

→ Time Complexity : $O(b^d)$.

* Alpha - Beta Pruning (α - β) :-

→ Cut off search by exploring less no. of nodes.
→ α → Max , β → Min.



→ Best and average case time complexity : $O(b^{d/2})$
→ Worst case time complexity : $O(b^d)$.

* Neural Networks :-

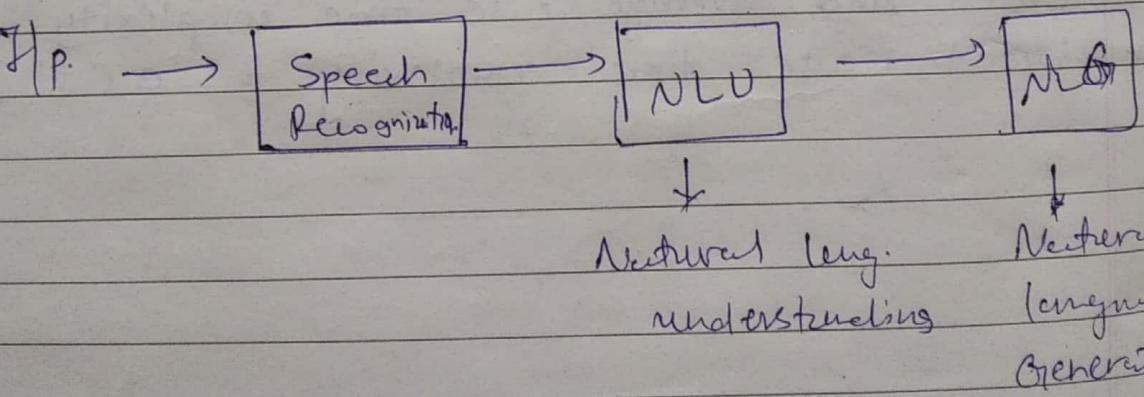
→ A neural network is a series of algorithms that endeavors to recognize underlying relationships in a set of data through a process that mimics the way the human brain operates.

* NLP : Natural language processing :-

→ How human communicate with each other.
→ ~~Complete stor~~ computer should replicate the same thing.

→ Application of NLP :-

- Speech recognization
- Sentiment analysis.
- Machine translatory
- Chat bot.



* NLU : What do the users say?
their intent ? Meaning ?

Challenges : lexical Ambiguity ~~(Structure)~~
Syntactic Ambiguity Structure
Meaning → Semantic Ambiguity ~~(Structure)~~
Praymetic Ambiguity

→ The (tank) was full of water.

→ Old men and women wer taken to saf plce.

→ The cun hit the pole while it was moving.

→ The police are coming.

* NLG : Natural Language Generation

What should we say to user ?

↳ It should be intelligent and conversational.
↳ Deal with structured data
↳ Text / sentence planning.

* Steps in NLP process.

1) Morphological Analysis.
2) Syntactic Analysis
3) Semantic Analyzer
4) Discource Integration : Referential ambiguity
5) Praymatic Analysis

* levels of syntactic analysis:

1) Part of speech (tagging)
2) Constituency parsing
3) Dependency parsing.

→ Lexical analysis focuses on smaller chunks
→ Semantic analysis focuses on larger chunks.

* Symantic analysis

lexical analysis
- Studying the meaning of individual word
- Studying the combination of individual words.

* Elements of symantic analysis:

1) ~~Hyponymy~~ → color : ~~Hypony~~ Hypernym
blue, yellow : Hyponyms.

2) Homonymy → One word diff. meaning.
Rose : flower
Rose : the past form of Rise!

3) Polysemy → lexical term that have the same spelling but multiple closely related meanings.
→ 'man' may mean the 'human species' or a 'male human' or an adult 'male human'.

4) Antonymy → rich/poor - moon/sun
5) Synonymy → author/writer - fate/destiny.

* Building blocks of semantic system:..

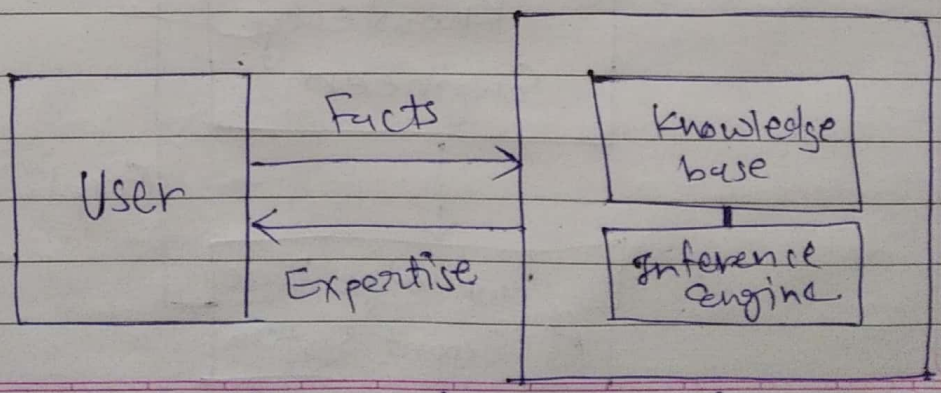
1) Entities 2) Concepts 3) Relations 4) Predicates

* The Text Analysis Process:..

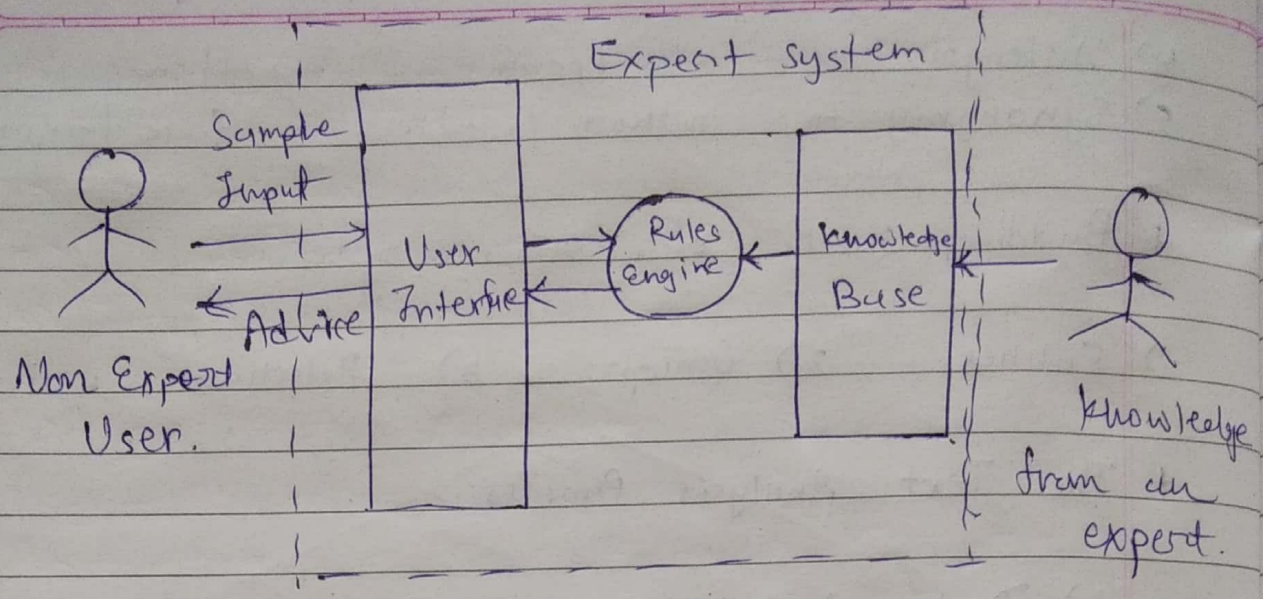
1) Data Acquisition
2) Data Preparation.
   - Text cleansing
   - Tokenization
   - Part of speech tagging
   - Parsing
3) Data Analysis
4) Data visualization

* Expert system main components:

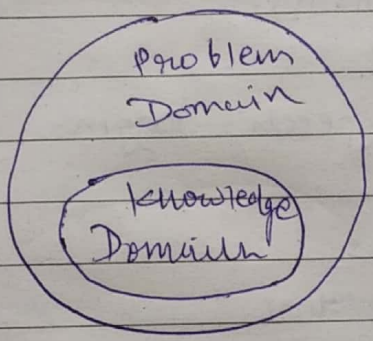
1) knowledge base : obtainable from books, knowledgable person, magazines etc...
2) Inference engine : draws conclusions from the knowledge base.



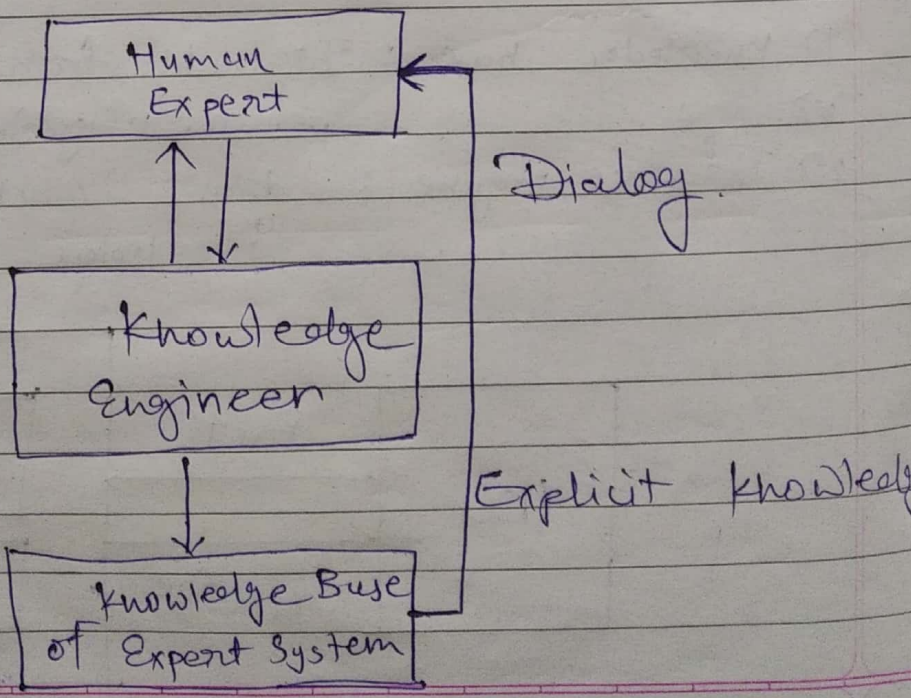
Basic functions of Expert system



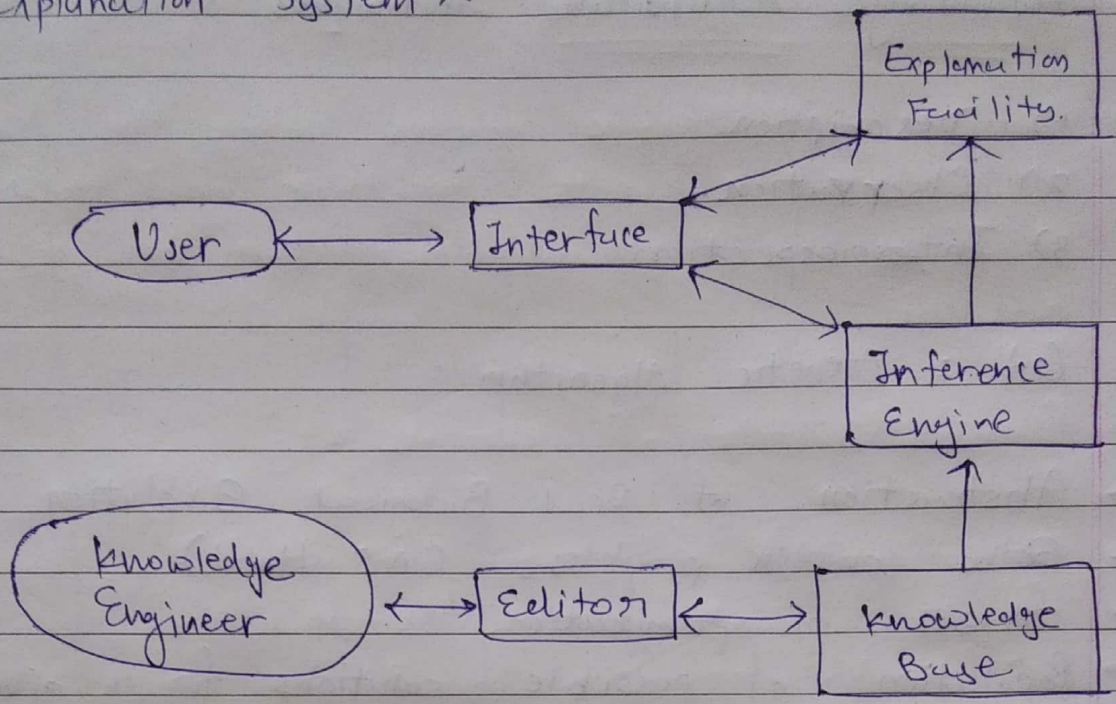
* Problem and knowledge Domain:-



* Development of Expert system:-



* Explanation System :-



* The Expert System Life-cycle :-

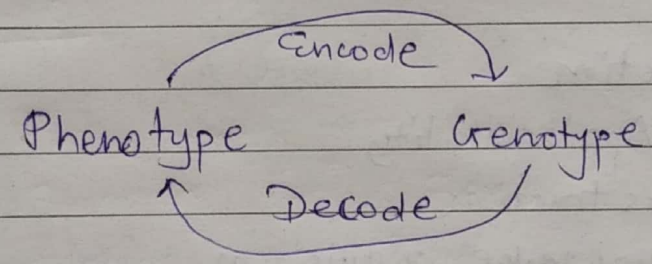
1) Problem Selection
   → Domain Applicability
2) Prototype construction
   → Initial knowledge acquisition
   → Basic problem approach.
   → Knowledge representation
   → Prototype implementing
   → Prototype testing
   → Prototype demonstration , ... etc.
3) Formalization
4) Implementation
   → Prototype revision
   → Internal integration
   → Internal verification
   → System framework development.
5) Evaluation
   → long term Evaluation.

* Knowledge Acquisition Techniques :-

1) Description
2) Observation
3) Introspection

* GA : Genetic Algorithm :-

→ Abstraction of Real Biological Evolution.
→ Solve complex problems (NP Hard)
→ focuses on optimization.
→ Population of possible solutions for a given problem.
→ From a group of individuals, the best will survive

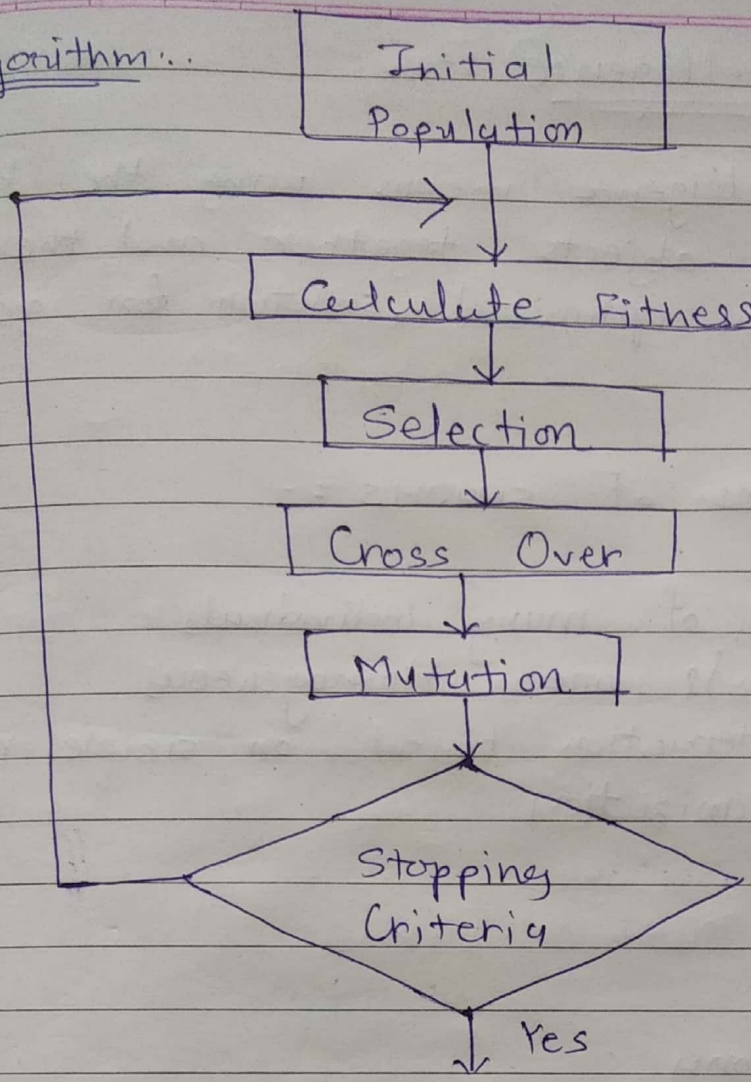


→ Population is a subset of solutions in the current generation.
→ Set of chromosomes.

* Knowledge Acquisition technique

1) Forward Chaining	2) Backward Chaining
→ Bottom-up	→ Top-down
→ data driven	→ goal driven
→ Ex :- predictions, etc..	→ dignosis of blood cancer, etc

* Algorithm..



* GA Crossover:..
  1) One point crossover
  2) Multipoint crossover
  3) Uniform crossover

* GA Mutation :
  1) Bit Flip mutation
  2) Swap mutation
  3) Scramble mutation
  4) Inversion mutation.

* Swarm Intelligence :-

→ Swarm intelligence means using the knowledge of collective objects together and then reaching the optimized solution for a given problem.

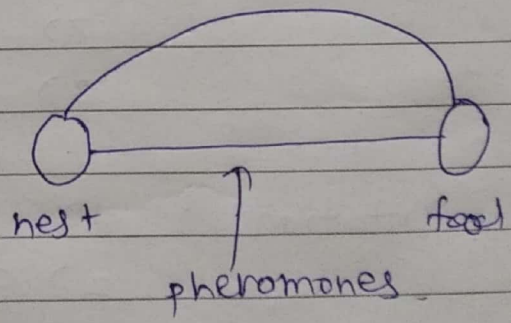
→ Characteristics of swarms :-

1) Composed of many individuals
2) Individuals are Homogeneous
3) local interaction based on simple rules
4) Self organization

→ Examples :

1) Ant colony
2) Swarm of Bees
3) flocks of Birds.

1) Ant colony :-



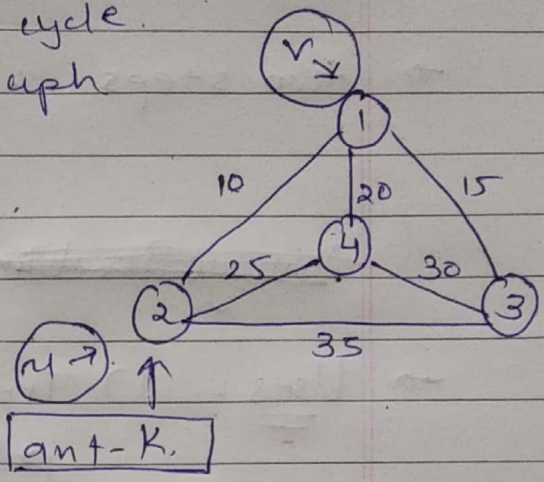
→ Ant colony optimization can be used to solve hard problems like TSP, Quadratic Assignment Problem (QAP).

ACO - TSP :-

→ Given a graph with n nodes, should give the shortest Hamiltonian cycle.

→ m ants traverse the graph

→ Suppose ant k is at u.



$N_k(u)$ be the nodes not visited by k.

T_{uv} be the pheromone trail of edge (u, v).

k jumps from u to a node v in $N_k(u)$ with probability,

$$P_{uv}(k) = T_{uv}\left(\frac{1}{d(u,v)}\right)$$

m ants are started at random nodes.
They traverse the graph prioritized on trails and edge weights.
An iteration ends when all the ants visit all nodes.
After each iteration, phepromone trails are updated.

New trails should have two components,
Old trail left after evaporation and
Trails added by ants traversing the edge during the iteration

$\therefore T_{uv} = (1-\gamma^0)\, T_{uv} + \text{Change In}\,(T_{uv})$

2) Swarm of Bees :

→ two steps : ManageBeeActivity() } *
CalculateVectors()

~~ManageBeeActiv~~

→ Uses : 1) Training neural networks for pattern recognition.
2) Forming manufacturing cells
3) Scheduling jobs for a production machine.
4) Data clustering