Page No.
Date: / /

AI

Artificial Intelligence
(man-made) (Thinking power)

Def^n: AI is a branch of computer science by which we can create intelligent machines, which can behave like a human, think like humans, & able to make decisions.

Foundations of AI

AI's basis is made on some fundamental ideas & elements that enable it to func^n in a variety of situations.

1) **Data & knowledge Representation**

The foundation of AI is data. An AI system needs huge amounts of data to find patterns & acquire insights to "learn" & make decisions. The goal of knowledge representation, is to organise this material such that computers can understand, store & retrieve it.

2) **Algorithms**

An AI system can assess data & draw conclusions from it because of algorithms, which are detailed inst^{n}s or set of rules. These

algorithms provide the basis of AI, allowing it to learn from data & make predictions.

3) Statistics & Mathematics

The foundation for understanding & evaluating data patterns is mathematics, especially statistics & calculus. Statistical techniques help in the development of models that show connections in data, enabling AI to make sensible choices.

4) Neuroscience & Cognitive Science

By providing blueprint of human learning, memory, & decision-making, cognitive science & neuroscience help build more "human-like" robots, which form the basis of AI.

History of AI

Maturation of AI [1943-1952]

Birth of AI [1952-1956]

Golden year - Early enthusiasm [1956-1974]

First AI winter [1974-1980]

Boom of AI [1980-1987]

Second Winter [1987-1993]

Page No.
Date : / /

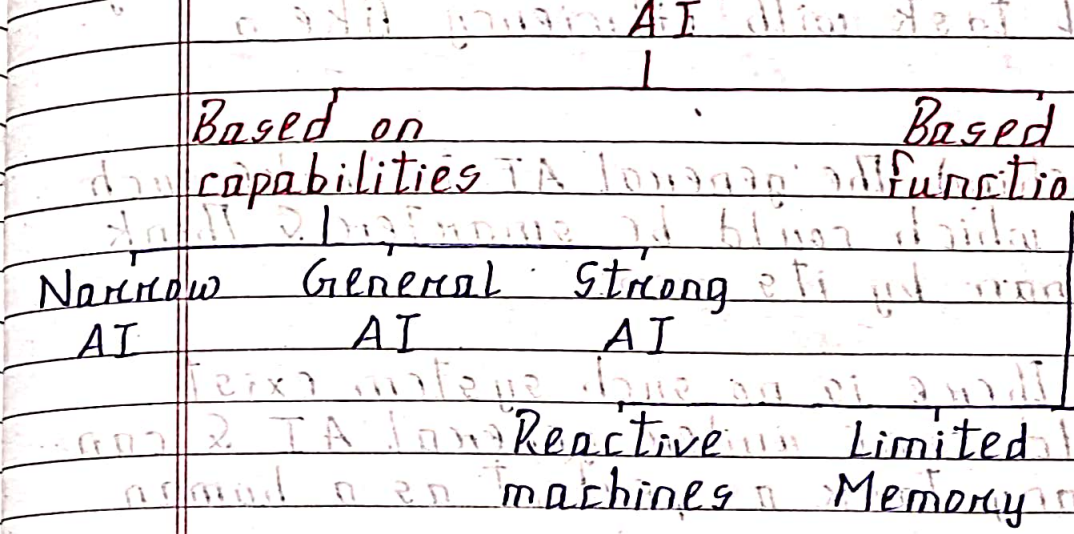
Emergence of intelligent agents [1993-2011]

Deep learning, big data [2011-present]

Application of AI

1) Astronomy
2) Healthcare
3) Gaming
4) Finance
5) Education
6) E-commerce
7) Robotics
8) Agriculture

Types of AI



Page No.
Date: / /

1) Weak OR Narrow AI

=> It is a type of AI which is able to perform a dedicated task with intelligence.

=> It is the most common & currently available AI.

=> Cannot perform beyond its field or limitations as it is trained for one specific task. Hence, it is also called Weak AI.

Eg. Apple Siriis

2) General AI

=> It is a type of AI which could perform any intellectual task with efficiency like a human.

=> The idea behind the general AI to make such a system which could be smarter & think like a human by its own.

=> Currently, there is no such system exist which could come under general AI & can perform any task as perfect as a human.

Page No.
Date: / /

3) Super AI

=> It is a type of intelligence of systems at which machines could surpass human intelligence, & can perform any task better than human with cognitive properties.

=> It is still a hypothetical concept of AI

4) Reactive Machines

=> Do not store memories or past experiences for future actions.

=> Only focus on current scenarios & react on it as per possible best action.

Eg. IBM's Deep Blue System
Google's AlphaGo

5) Limited Memory

=> Can store past experiences OR some data for a short period of time.

Eg. Self-driving cars

Page No.
Date: / /

6) Theory of Mind

=> Should understand the human emotions, people, beliefs, & be able to interact socially like humans.

=> This type of AI machines are still not developed.

7) Self-Awareness

=> This is the future of AI.

=> These machines will be super intelligent, & will have their own consciousness, sentiments & self-awareness.

=> These machines will be smarter than human mind.

Page No
Date / /

Difference bet^n Strong & Weak AI

	STRONG AI	WEAK AI
1	AI with human like intelligence.	AI designed for specific tasks
2	Thinks, understands & learns like a human.	Performed pre-defined tasks efficiently.
3	Has conciousness & self-awareness.	Lacks conciousness & self awareness
	Eg. Siri, Alexa, Chatbot	

Advantage of AI

→ Do not get tired OR wear out easily.
→ Rational decision maker.
→ Calculate with precision without break.

Disadvantage of AI

→ Incures high cost
→ May occur errors.

Page No.
Date: / /

AI is classified into 4 categories :-

1) Think like human
AI should process info like human brain. i.e how humans think, solve problems & make decisions.

2) Think rationally
AI should follow strict logic & reasoning, even if it does not think exactly like a human

3) Act like Human
AI behaving like a human, even if it does not actually like one by using techniques like ML & deep learning to mimic human actions.

4) Act Rationally
AI does not need to act like a human but should act in a way that is more effective for achieving its goal.

Temp. management in a room → Act like human

Automated Vaccum cleaner → Act like human

Tic Tac Toe gaming → Think rationally

Colour Detection → Think rationally

Self Tuition → Act rationally

Face Recognition → Think like human

Automated Sliding Door → Act like human

Part picking robot → Act rationally

Patient Diagnostic Management → Act rationally

Self Driving Car → Act rationally

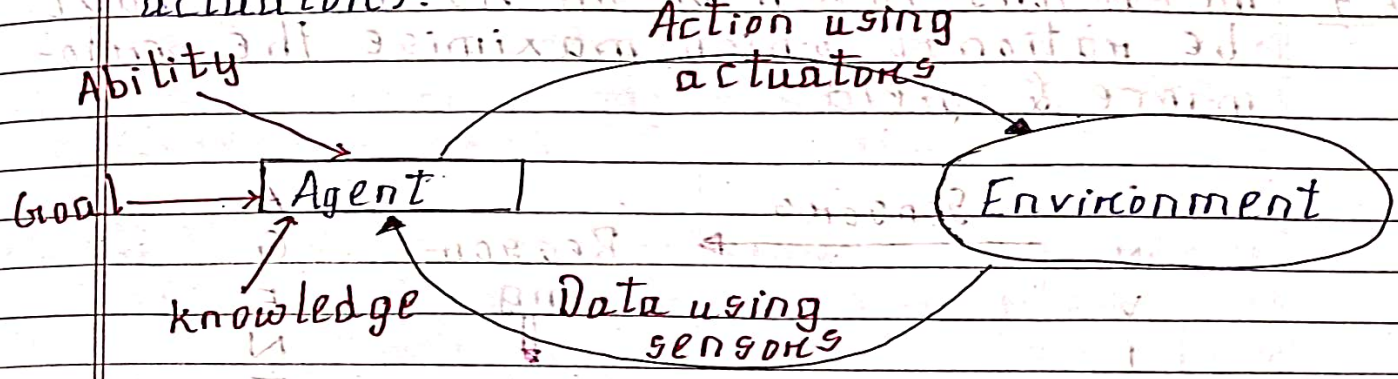
Agents

⇒ An AI system is composed of an agent & its environment. The agents act in their environment.

⇒ An AI agent is anything that can be viewed as :-

→ Perceiving its environment through sensors.

→ Acting upon that environment through actuators.



Page No.
Date: / /

Structure of an AI agent

Agent = Architecture + Agent Program
(machinery that agent executes on)

Defⁿ :- An agent is an independent program OR entity that interacts with the environment by perceiving it's surroundings via sensors then acting through actuators.

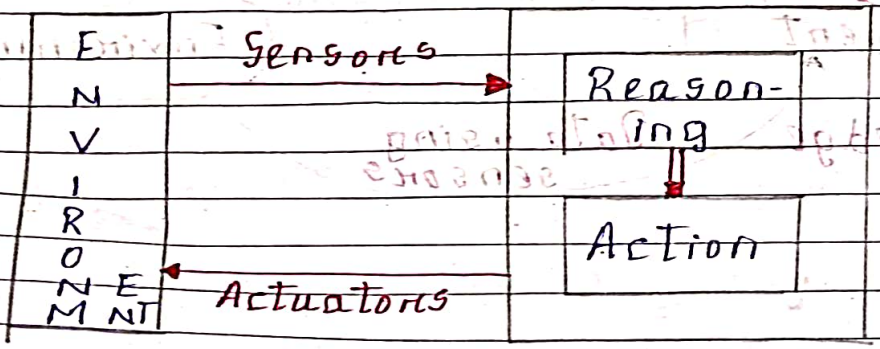
4 Rules must followed by AI Agents

RULE 1:- An agent must be able to perceive the environment.

RULE 2:- The environmental observation must be used to make decisions.

RULE 3:- The decisions should result in action.

RULE 4:- The actions taken by an AI agent must be rational which maximise the performance & yield.



Page No.
Date: / /

Rational Agent

=> It is agent that performs actions to achieve the best possible outcome based on its perceptions & knowledge.

Factors on which Rationality Depend

→ Agent's prior knowledge regarding the environment.
→ Actions that agent can perform.
→ Agent percept sequence till the moment.

Requirements

→ Information gathering capability.
→ Learning ability.
→ Ability to adapt
→ An agent is autonomous.

Intelligent Agent

=> It is agent that perceives its environment, processes information & takes actions to achieve a goal.

=> It operates autonomously & can adapt its behaviour based on experience.

Page No.
Date: / /

Factors to design Intelligent Agents

Percepts : input to the systems

Actions : The output of our system

Goal : what the agent is expected to achieve.

Environment : what the agent is interacting with.

Difference bet^n Rational and Intelligent Agent

	INTELLIGENT AGENT	RATIONAL AGENT
1	Perceive its environment & take actions to achieve specific goals.	Decision based on logical reasoning & optimize it's behaviour to achieve goal.
2	Perceive it's environment through sensors.	Based on information available to it.
3	Makes decision based on set of rules.	Decision based on logical reasoning.
4	Operates independently of human intervention.	Runs independently but based on logical reasoning.
	Eg. self driving car	Eg. Logistic planning

Page No.
Date : / /

Agents

- Simple Reflex
- Model Based
- Goal Based
- Utility Based
- Learning based

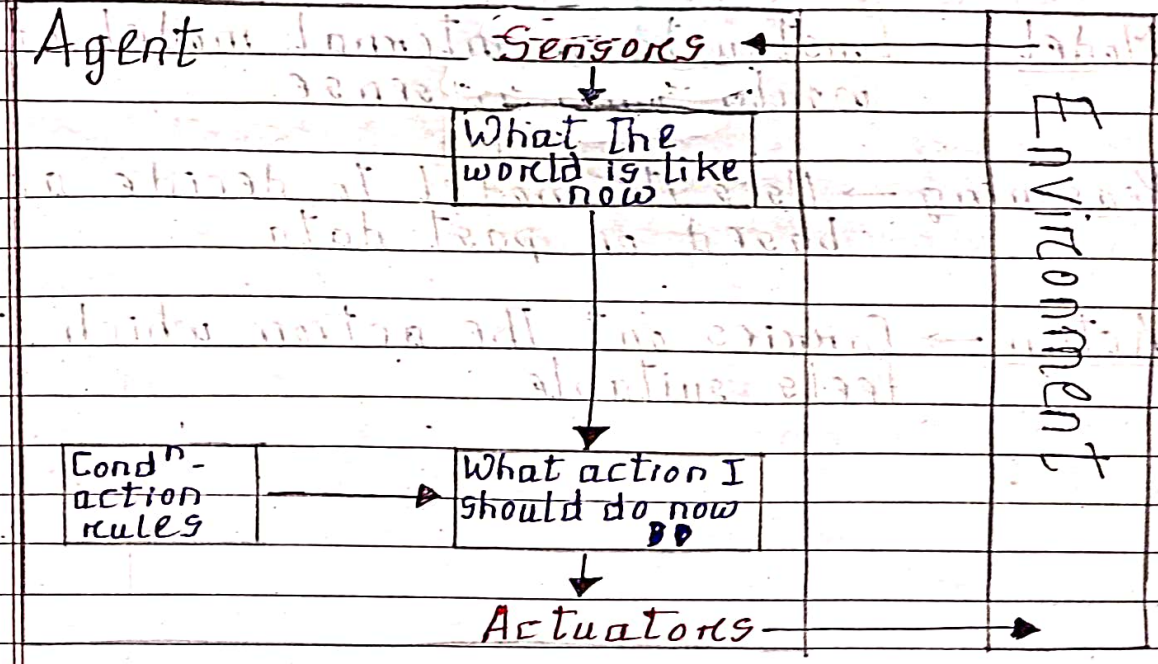
1) Simple Reflex Agents

=> It allows pre-defined rules to make decision.

=> It respond to current situation without considering past OR future.

=> Suitable for stable rules & straight-forward actions as its behaviour is purely reactive & responsive to immediate environment changes.

Eg. Automated door, Room Temperature



Page No.
Date: / /

Advantages : i) Easy to design
ii) Require minimal resources
iii) Real time response
iv) Highly reliable in situation when sensor inputs are accurate.

Disadvantages : i) Prone to errors if sensors are faulty
ii) No memory / states.

2) Model Based Agents

=> Action is based on current percept & internal state representing & observed world.

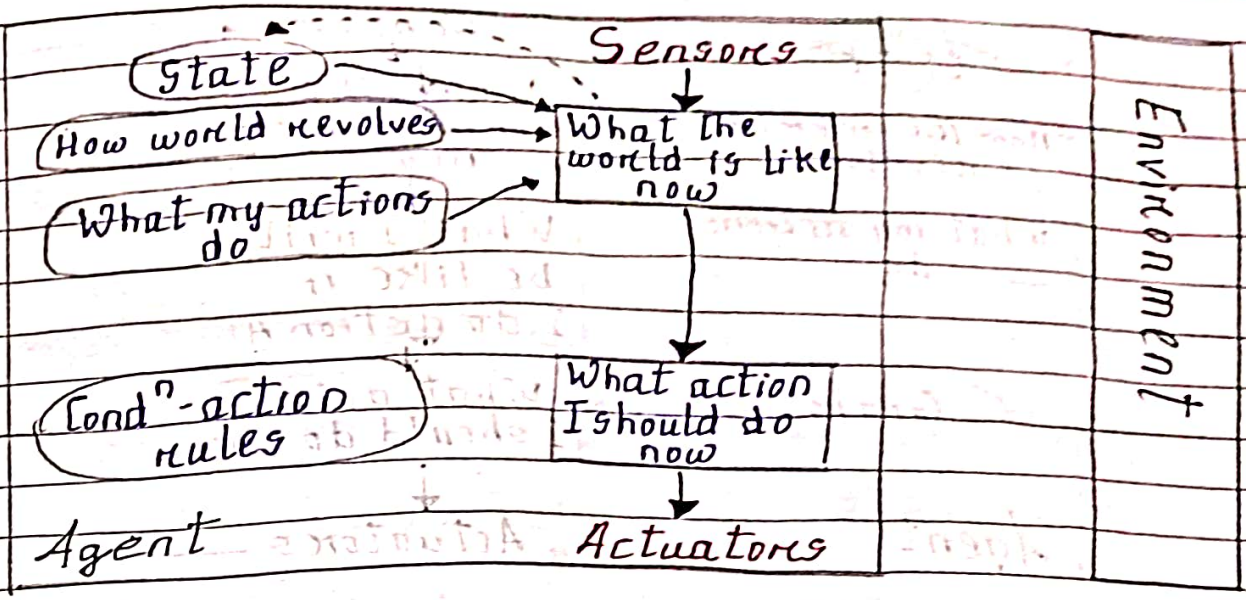
4 stages :-

i) Sense → Perceive current state

ii) Model → Constructs an internal model of the world what it sense.

iii) Reasoning → Uses its model to decide action based on past data.

iv) Action → Carries out the action which it feels suitable



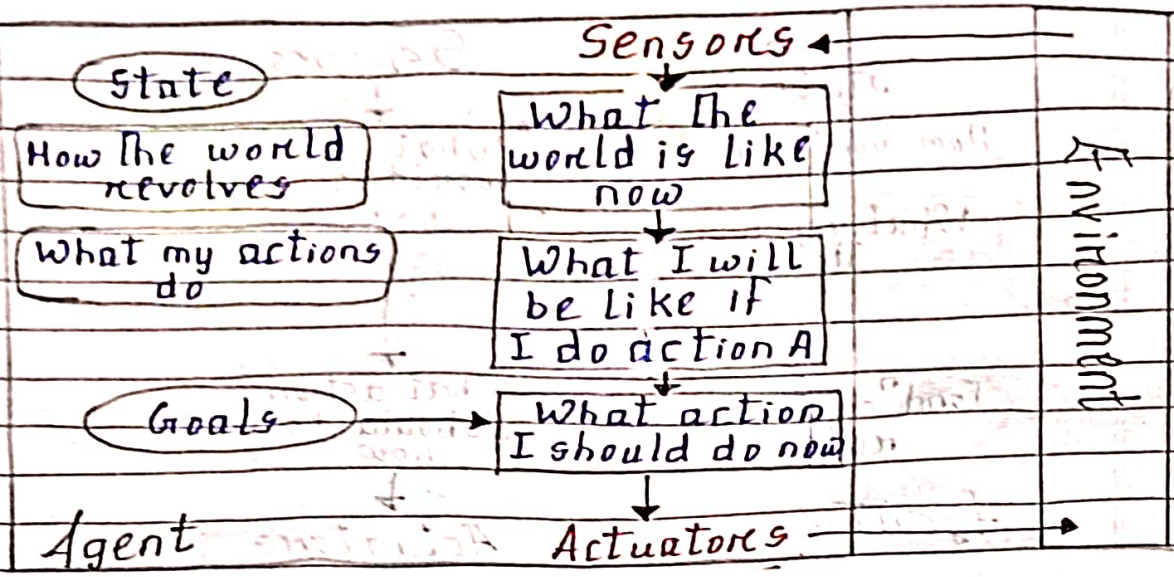
Advantages :- i) quick & efficient decision making
ii) Adaptability to change the environment.

Disadvantages :- i) building & maintenance is expensive
ii) Model may miss real world strategy

3) Goal Based Agents

=> It uses information from the environment to achieve specific goal.

=> Have 5 steps :- i) Perception
ii) Reasoning
iii) Action.
iv) Evaluation
v) Goal Completion



Advantages :- i) Simple to implement & understand
ii) Easy to evaluate

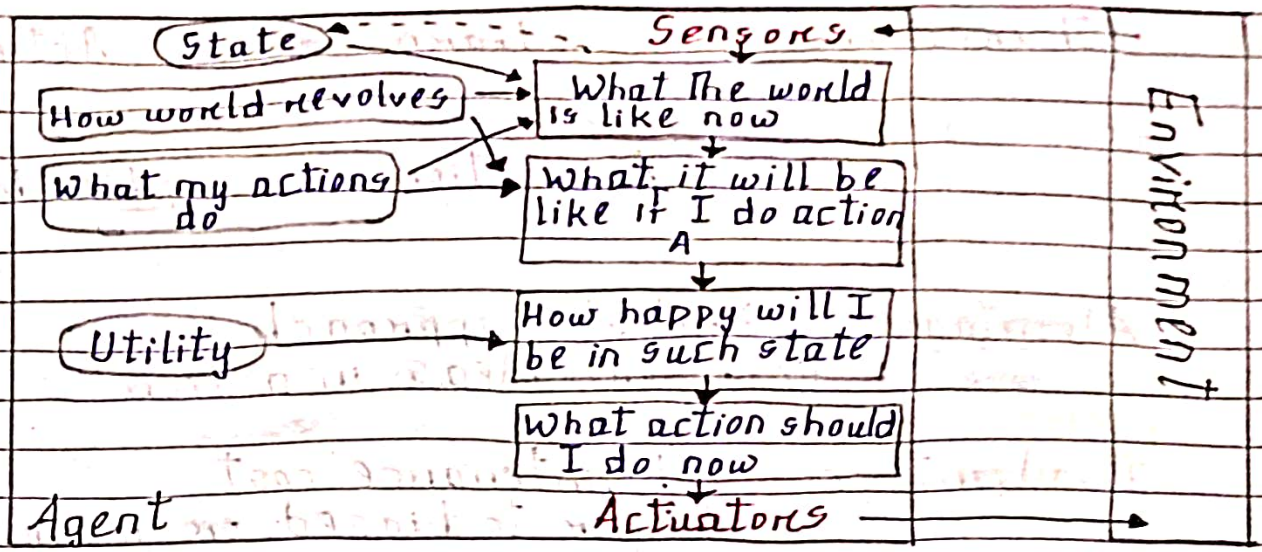
Disadvantages :- i) limited to specific goal
ii) Unable to adapt with changes.

4) Utility Based Agent

=> These are AI agents that make decisions based on maximizing a utility function or value

=> They choose the action with the highest expected utility which measured that how good the outcome is.

=> Help to deal with complex & uncertain situations with flexibility & adaptability.



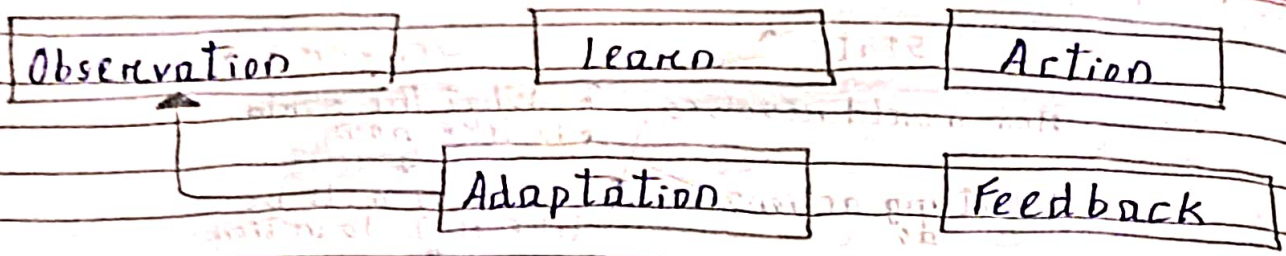
Advantages :- i) Handle wide range of decisions.
ii) Learn from experience & adjust their decision making strategy.

Disadvantages :- i) Expensive.
ii) Difficult for humans to understand.

5) Learning Based Agent

=> These are AI agents that can learn from past experiences & improve it's performance & it acts with basic knowledge & adapts automatically through machine learning.

=> Consists of :- i) Learning element
ii) Critic
iii) Performance element
iv) Problem solving



Advantages : i) Realistic approach
ii) Convert ideas in action

Disadvantages : i) Maintenance cost
ii) Prone to biased or incorrect decision.

PEAS Representation :

P → Performance
E → Environment
A → Actuators
S → Sensors

1) English Tuition System

P → Student improvement in language proficiency, accuracy, fluency

E → Classroom, online platform, students, teaching materials.

A → Teacher's speech, assignments, quizzes

S → Microphone, keyboard, student responses.

Page No.
Date : / /

2) Patient Diagnostic System

P → Accurate diagnosis, Min cost, patient recovery rate

E → Hospital, clinic, patient data, medical record

A → Generate reports, Display diagnosis

S → Patient symptoms, Medical reports

3) Vaccum Cleaner

P → Cleanliness, Battery efficiency, Time.

E → Home, office, dust, dirt, obstacle like furniture

A → Wheels, brushes, suction mechanism

S → Dirt sensors, bump sensors, battery level indicator, obstacle sensors.

4) Self Driving Car

P → Safe, fuel efficiency, adherence to traffic rules

E → Roads, Traffic, Weather cond's, pedestrians

A → Steering, brakes, indicators, wipers

S → Cameras, GPS, speedometer

Page No.
Date: / /

5) Part Picking Robot

P → Speed, Error minimization, efficiency

E → Factory Floor, storage bins, diff. parts

A → Robotic arms, grippers, conveyer belts

S → Cameras, barcode scanners, force sensors.

6) Refinery Controller

P → Purity, yield, safety

E → Refinery, raw materials

A → Valves, pumps, heaters, stirrers

S → Temp, flow, pressure

7) Satellite Image Analysis System

P → Terrain, correct categorisation of objects.

E → Conveyer belt with parts, bins.

A → Labelling system, categorisation tags

S → Satellite images, image sensors, cameras

Page No.
Date: / /

Environment

The environment is where agent lives, operate & provide the agent with something to sense & act upon it.

Fully Observable VS Partially Observable

	FULLY OBSERVABLE	PARTIALLY OBSERVABLE
1	Complete information about the environment.	Limited OR incomplete information
2	Easier decision making as no hidden variables exist	Harder as the agent must infer missing details.
3	No uncertainty; all factors are known	High uncertainty due to missing data.
	Eg. Chess, Tic-Tac-Toe	Eg. Self Driving Car

Deterministic Vs Stochastic

	DETERMINISTIC	STOCHASTIC
1	Next state is determined by the current state & action.	Next state has uncertainty; multiple outcomes are possible.

2	No randomness	Uses probability to determine outcomes.
3	Not needed as results are fixed.	Require probabilistic reasoning.
	Eg. Chess, solving math eq^{n}	Eg. Rolling a dice, self driving cars.

Episodic Vs Sequential

	EPISODIC	SEQUENTIAL
1	Each action is independent; past actions do not affect future ones.	Each action influences future decisions & outcomes.
2	No memory is needed	Memory is essential to track previous states & actions
3	Only focuses on immediate rewards.	Requires long term planning & strategy.
4	Eg. Spam email filtering	Eg. Chess, self driving cars.

Page No.
Date

Static Vs Dynamic

	STATIC	DYNAMIC
1	Remains unchanged untill the agent acts.	Can change on its own, even if the agent does nothing.
2	Simpler: agent doesn't need to track time dependent changes.	More complex, as agent must adapt to changes.
3	Not required; a single observation is sufficient.	Required to keep track of environmental changes.
	Eg. Puzzle solving	Eg. Stock Market

Discrete Vs Continuous

	DISCRETE	CONTINUOUS
1	Finite no. of distinct states.	Infinite or highly granular states
2	Limited set of possible actions.	Actions can take any value within range.
3	Easier to model & solve.	More complex.
	Eg. Chess, Tic Tac Toe	Eg. Robot arm movement

FO → Fully observable
PO → Partially "
D → Deterministic

SA → Single Agent
MA → Multi "
S → Sequential

Page No.
Date / /

1) Crossword Puzzle → FO, SA, D, S, Static, Discrete

2) Chess with a clock → FO, MA, D, S, Static, Discrete

3) Poker → PO, MA, Stochastic, S, Static, Discrete

4) Medical Diagnosis → PO, SA, Stochastic, S, Dynamic, cts

5) Refinery Controller → PO, SA, Stochastic, S, Dynamic, cts

Page No.
Date: / /

Searching

- Informed / Heuristic
  - A* search
  - Hill climbing
  - Best First Search
- Uninformed / Blind
  - BFS
  - DFS
  - DLS
  - UCS

Blind Search

=> It has no additional informations about the states that provided in the problem definition

=> Doesn't take ~~account~~ into account either execution efficiency or planning efficiency.

=> Doesn't contain information about domain except information given by the definition of problem

Heuristic Search

=> In this, we explore the node or state i.e. most likely to be nearest to the goal state.

=> A heuristic func^n has some knowledge about the problem so that it can judge how the current state is nearer to goal state.

Page No.
Date: / /

Problem Solving Using State Space

1) **Initial State** : describe possible situations from where the problem solving begins.

2) **A set of actions** : Describes the possible actions. Represented by SF(x)
initial + SF → state
state → space

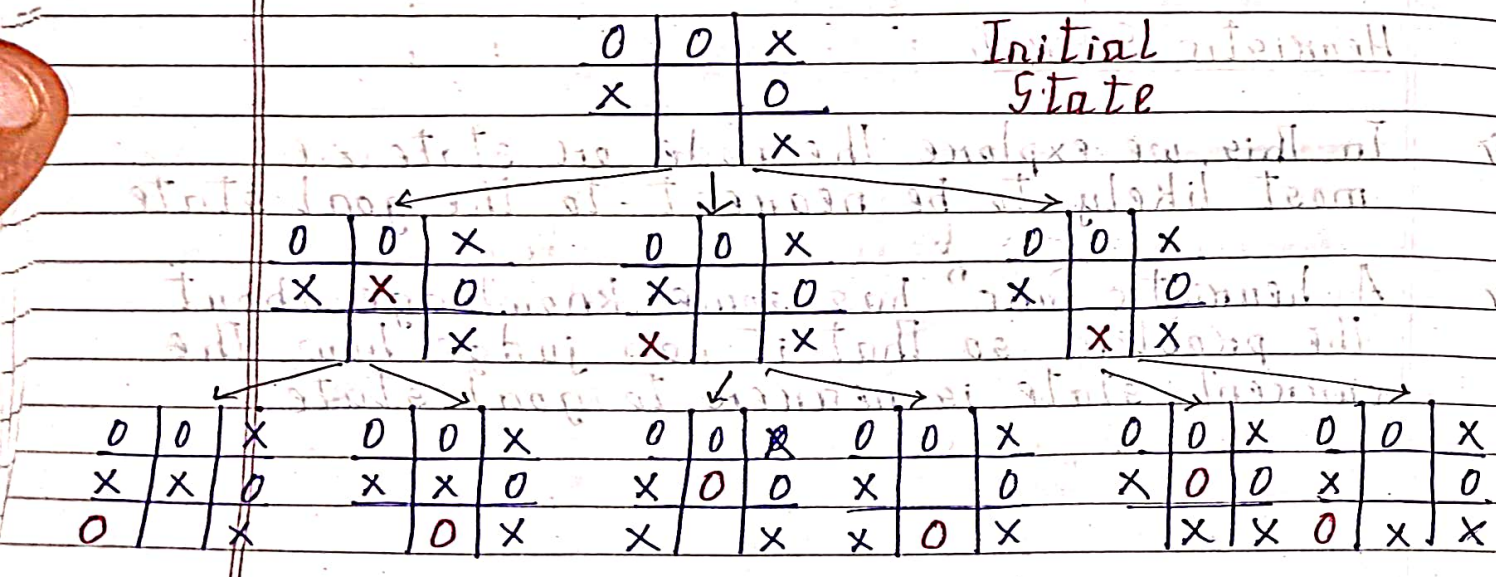
3) **State Space** : Possible configurations of the relevant objects associated with problem.

4) **Goal Test** : Represents acceptable sol^n

5) **Path cost Func^n** : Numeric cost to each path.

Case Study Problems

1) **Tic-Tac-Toe Problem**



O	O	X
X	X	O
O	X	X

Draw

O	O	X
X	X	O
X	O	X

Win

O	O	X
X	X	O
X	O	X

Win

O	O	X
X	O	O
X	X	X

Win

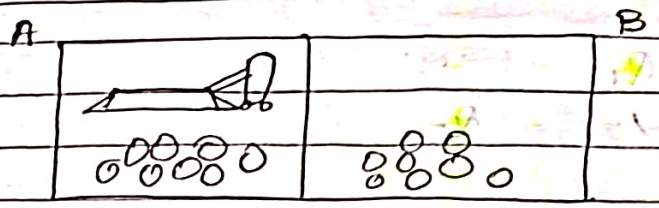
O	O	X
X	O	O
X	X	X

Win

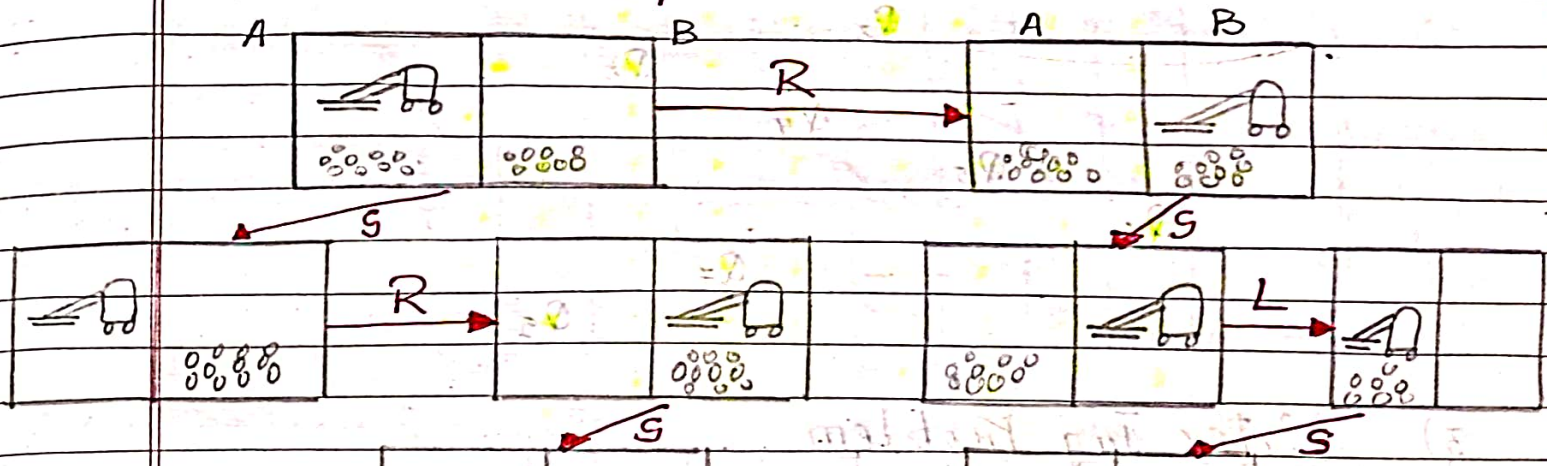
O	O	X
X	X	O
O	X	X

Draw

2) Vaccum Cleaner



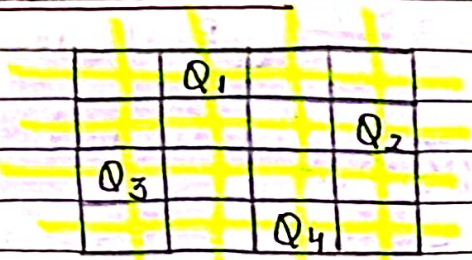
state → integer dirt & robot location
action → Left, right, suck
goal test → no dirt at A & B
path cost → 1 per action



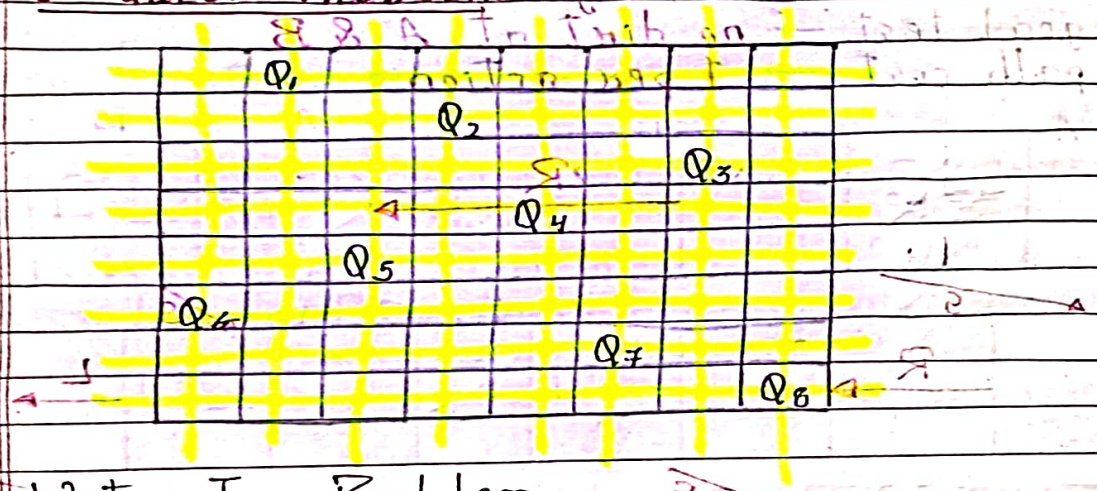
3) N-Queen Problem.

Each row & each column should contain exactly one queen in them.

4-Queen Problem

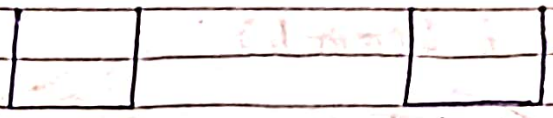


8-Queen Problem



3) Water Jug Problem

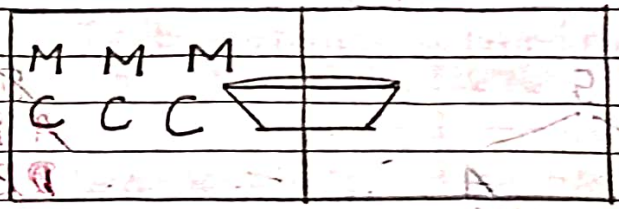
2 container consists 4L & 3L of water respectively. Neither has any measuring markers on it. There is a pump that can be used to fill the jugs with water. One has to get exactly 2 litres of water in 4L jug.



<0, 0> // initial state
<4, 0> // put 4L in x
<1, 3> // transfer 2L to y
<1, 0> // remove 3L
<0, 1> // alternate x & y
<4, 1> // put 4L in x
<2, 3> // transfer 2L to y
<2, 0> // drain y

4) Missionaries & Cannibals Problem

Cond°:
C ≤ M



3 Missionaries
3 Cannibal
2 River Banks

<1, 3, 3>	<2, 0, 0> // initial state
<1, 3, 1>	<2, 0, 2>
<1, 3, 2>	<2, 0, 1>
<1, 3, 0>	<2, 0, 3>
<1, 3, 1>	<2, 0, 2>
<1, 2, 0>	<2, 0, 3>
<1, 2, 2>	<2, 1, 1>
<1, 0, 2>	<2, 3, 1>
<1, 0, 3>	<2, 3, 0>
<1, 0 1>	<2, 3, 2>
<1, 0, 2>	<2, 3, 1>
<1, 0, 0>	<2, 3, 3> // goal state

Page No.
Date: / /

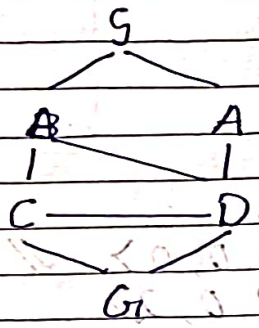
BFS (Breadth First Search)

=> FIFO queue is used.

=> Level to level expansion

Time Complexity $\longrightarrow O(b^d)$

Space Complexity $\longrightarrow O(b^d)$



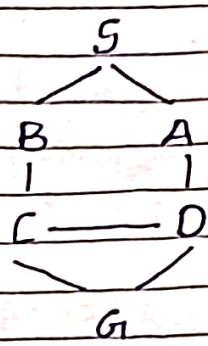
S
A B
D B D
Ø C
Ø G
G

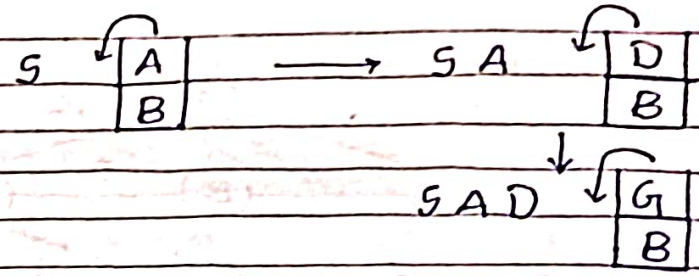
$S \rightarrow A \rightarrow B \rightarrow D \rightarrow C \rightarrow G$

DFS (Depth First Search)

=> Stack is used

=> Depth expansion





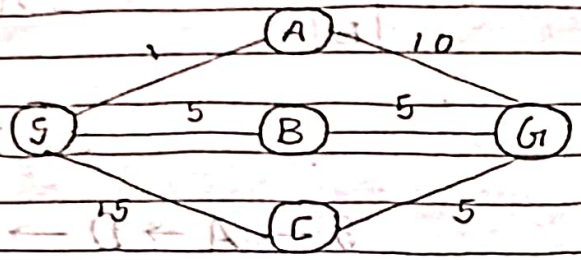
$S \rightarrow A \rightarrow D \rightarrow G$

BFS Vs DFS

	BFS	DFS
1	Traverse the tree level wise	Traverse the tree depth wise.
2	Implemented using queue.	Implemented using stack.
3	No backtracking is allowed.	Backtracking is allowed.
4	Never get trapped into infinite loops	Get trapped into infinite loops.

UCS (Uniform Cost Search)

=> Expands the least node first, ensuring that the path to the goal node has the minimum cost.



{ S_0 }
S_0 { A_1, B_5, C_{15} }
A_1 { B_5, C_{15}, G_{10} } → S → A → G = 11
B_5 { C_{15}, G_{11}, G_{10} }
G_{10} { C_{15} }
C_{15} { G_{20} }

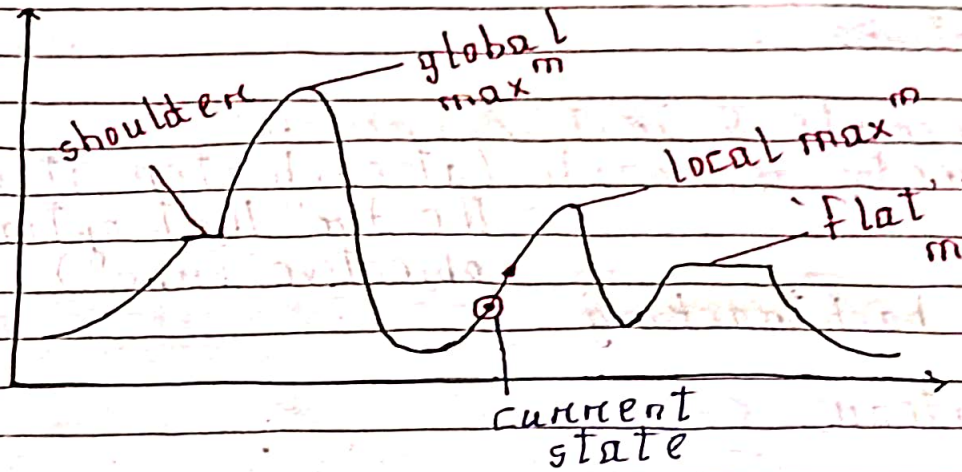
Visited nodes S_0 A_1 B_5 G_{10} C_{15}

Shortest path → S_0 B_5 G_{10}

Hill Climbing

⇒ Starts with a non-optimal state & iteratively improves its state until some predefined cond^n is met. The cond^n to be met is based on the heuristic func^n.

⇒ The starting point which is non-optimal state is referenced to as the base of the hill & it constantly iterate (climb) until it reaches the peak value, that is why it is Hill climbing.



Local Max^m :- It is a state better than its neighbours but not the best overall. While its objective func^n value is higher than nearby states, a global max^m may still exist.

Global Max^m :- It is the best state in the state diagram, where the objective func^n achieves its highest value. This is the optimal sol^n the algorithm seeks.

Flat local Max^m :- A region depicted by a st. line where all neighbouring states have the same value so every node is local max^m over the region.

Current State :- Agent is present currently.

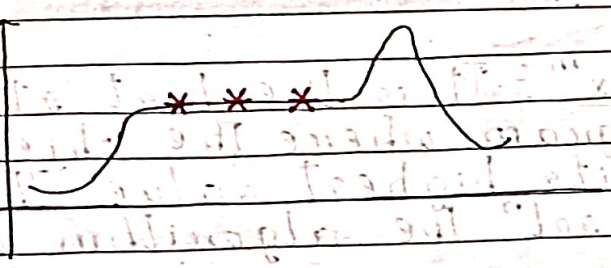
Page No.
Date: / /

Features

→ Generate & Test Approach
→ Greedy approach (at each step, it moves in the dir^n that optimizes the objective func^n)
→ No backtracking.

Plateau

It is a plate where the algorithm does not find any replacement that may lead the algorithm to wander aimlessly without further progress.



Ridges

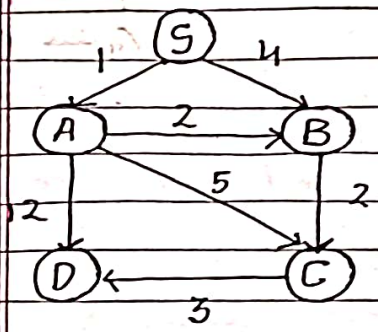
Because all dir^{n}s of movement are downward, every location on the ridge can be seen as a summit. Thus, in this circumstance, the algorithm fails.

Page No.
Date: / /

A* search Algorithm

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

- $f(n)$ → Total cost
- $g(n)$ → The cost to reach the next node
- $h(n)$ → heuristic value



	h(n)
S	7
A	6
B	2
C	1
D	0

S 0 + 7 = 7

S → A 6 + 1 = 7 X

S → B 4 + 2 = 6 X

S → B → C 4 + 2 + 1 = 7 X

S → A → B 1 + 2 + 2 = 5

S → A → B → C 1 + 2 + 2 + 1 = 6

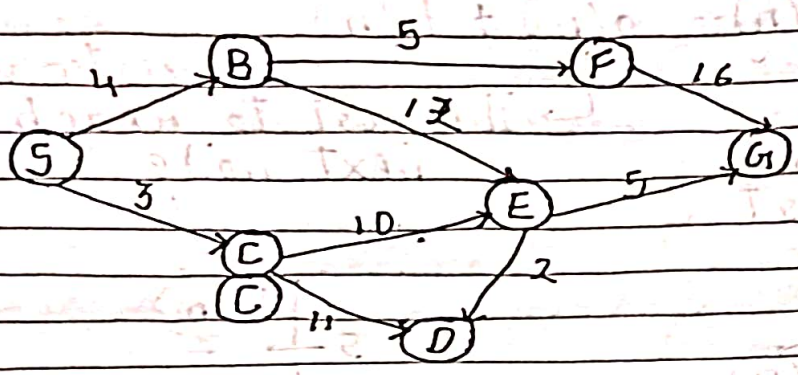
S → A → C 1 + 5 + 1 = 7 X

S → A → D 1 + 12 + 0 = 13 X

S → B → C 4 + 2 + 1 = 7 X

Page No.
Date: / /

S → A → B → C → D



S → 14
B → 12
C → 11
D → 6
E → 4
F → 11
G → 0

f(S) = 0 + 14 = 14
f(S-B) = 4 + 12 = 16
f(S-C) = 3 + 11 = 14
f(SC-D) = 3 + 11 + 6 = 20
f(SC-E) = 3 + 10 + 4 = 17 ✓
f(SB-E) = ~~4 + 5 + 4~~ = 4 + 12 + 4 = 20
f(SB-F) = 4 + 5 + 11 = 20

f(SCE-G) = 18

Page No.
Date : / /

Constraint Satisfaction Problem (CSP)

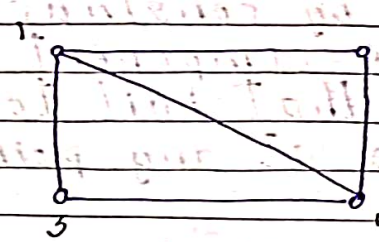
=> It involves a set of variables each of which has a domain of possible values & a set of constraints that defined allowable comb^n of value for the variable.

=> The goal is to find the value for each variable such that the constraint are satisfied.

A CSP is defined (X, D, C)

$X = X_1, X_2, \dots X_n$ ~~Doma~~ Variable
$D = D_1, D_2, \dots D_n$ Domain
$C = C_1, C_2, \dots C_n$ Constraint

Map Colouring Problem



$X = \{1, 2, 3, 4\}$
$D = \{$ Red, Green, Blue $)$
$C = \{$ adjacent nodes should not have same colour $\}$

Page No.
Date: / /

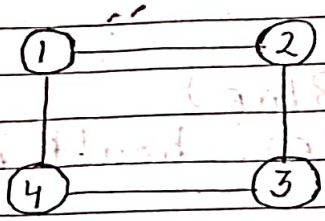
	1	2	3	4
Initial	R, G, B	R, G, B	R, G, B	R, G, B
1 = R	R	G, B	G, B	G, B
2 = G	R	G	G, B	B
3 = G	R	G	G	B

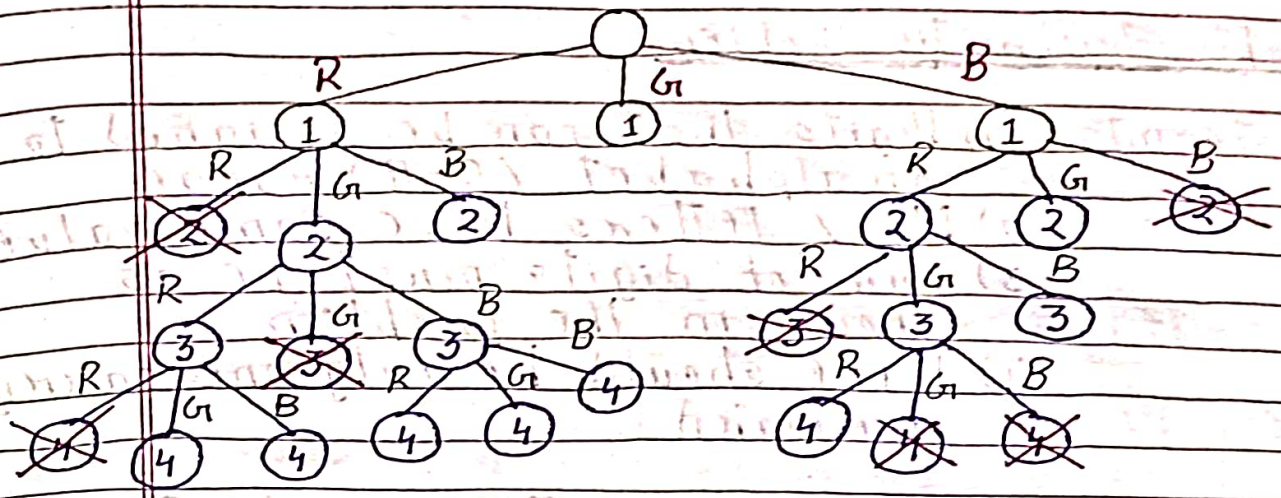
If we put 3 = B, then there would be 2 adjacent nodes having same colour which would be ERROR.

Backtracking

=> It is a general algorithm for solving CSP.

=> It accomplished this by constructing a sol^n implemently one component at a time discarding any sol^n that fail to satisfy the problem criteria at any point of times.





Heuristic Func^n Vs Heuristic Search

HEURISTIC FUNC^N	HEURISTIC SEARCH
1 Estimate cost of cheapest path from node n to a goal node.	It is a guided search
2 It is a way to inform the search about the dir^n to a goal.	It is a searching method that might not always find the best sol^n but it guaranteed to find a good sol^n in a regionable time.
3 It provides an inform way to guess which neighbour of a node will lead to a goal.	Helpful for a problem which take infinite time.

Page No.
Date : / /

* *

Crypto Arithmetic Problem

Constraints : 1) Digits that can be assigned to an alphabet (0-9) range
2) No 2 letters have same value.
3) Sum of digits must be as shown in the problem.
4) There should be only one carry forward

Q-
```
  T O
+ G O
-----
O U T
```

Left most digit = 1 & ≠ 0

Letter	Digit
T	→ 2
O	→ 1
G	→ 8
U	→ 0

```
  2  1
+ 8  1
-------
1  0  2
```

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

letter	Digit
S	→ 9
M	→ 1
N	→ 6
O	→ 0
R	→ 8
D	→
E	→ 5
Y	→

C_4 C_3 $C_2 = 1$ $C_1 = 1$

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

S + M, if m = 1, S + M ≥ 10
E + O, if O = 0, $C_2 = 1$

Q- EAT + THAT = APPLE

Letter		Digit
A	→	1
E	→	8
H	→	2
I	→	3
P	→	0
T	→	9

$C_3 = 1$ C_2 $C_1 = 1$

		E	A	T
		8	1	9
+	T	H	A	T
	9	2	1	9
A	P	P	L	E
1	0	0	3	8

Q- SOME + TIME = SPENT

Letter		Digit
S	→	1
O	→	9
M	→	3
E	→	4
T	→	8
I	→	5
P	→	0
N	→	6

C_4 $C_3 = 1$ C_2 C_1

		S	O	M	E
		1	9	3	4
+		8	5	3	4
1		0	4	6	8

Page No.
Date: / /

Q- BASE
BALL
GAMES

C = 1

~~7~~	4	8	3	
~~7~~	4	5	5	
1	2	9	3	8

Letter		Digit
B	→	7
A	→	4
S	→	8
E	→	3
L	→	5
G	→	1
M	→	9

Q- CROSS
ROADS
DANGER

	9	6	2	3	3
+	6	2	5	1	3
1	5	8	7	4	6

Letter		Digits
C	→	9
R	→	6
O	→	2
S	→	3
A	→	5
D	→	1
N	→	8
G	→	7
E	→	4

Page No.
Date: / /

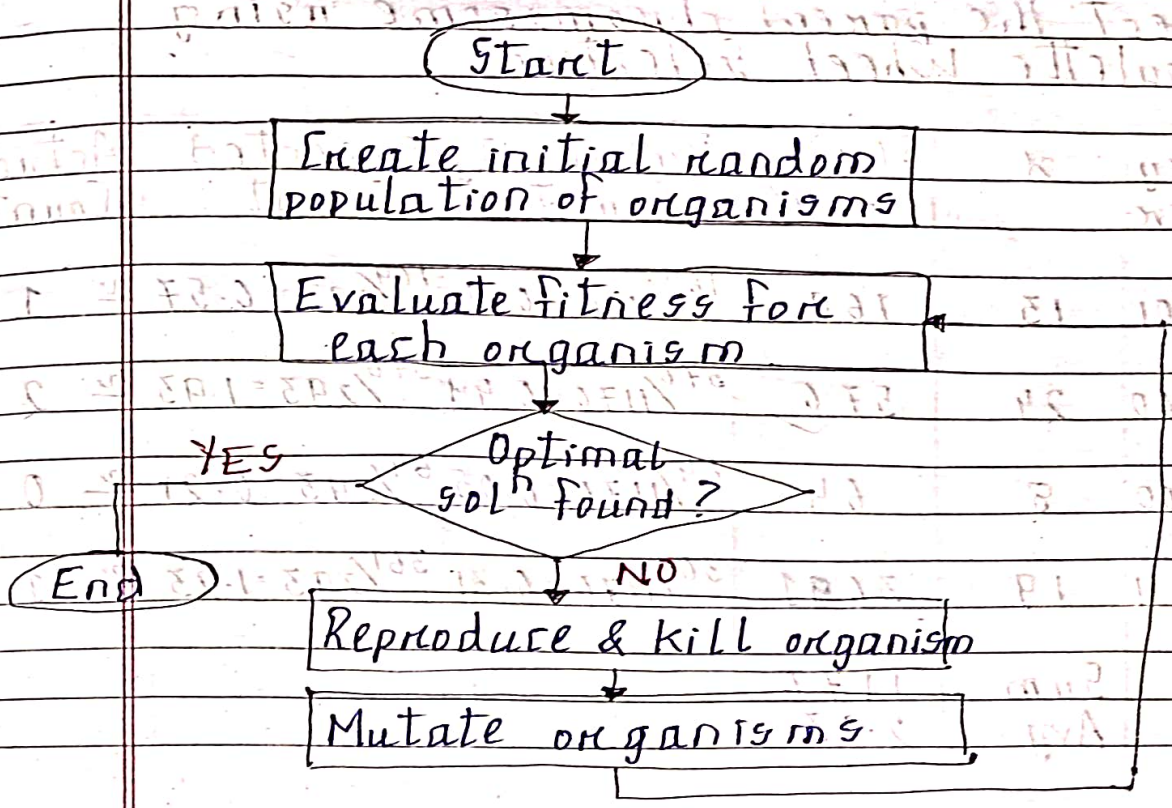
Genetic Algorithm

=> It is a search based optimization technique based on the principal of genetics & natural selection.

=> Developed by John Holland.

=> In GA, we have a population of possible sol^n to the given problem.

=> Each individual is assigned a fitness value f(x) & the fitter sol^n or value are given a chance to mate & yield more fitter individual that is known as survival of the fittest.



Page No.
Date: / /

Solving a GA Problem

Q- Maximise $f(x) = x^2$ where $x \in [0, 31]$ using binary encoding & Roulette wheel selection method.

A- **STEP 1** Selection of binary encoder with 6-bit string

$0 \longrightarrow 000000$

$31 \longrightarrow 111111$

STEP 2 Initialize the population size i.e. $n = 4$.

STEP 3 Randomly choose any 4 elements
Let $\{13, 24, 8, 19\}$

STEP 4 Select the parent chromosome using Roulette Wheel Selection

String no.	Binary encoder	x	$f(x) = x^2$	Probability Count	Expected Count	Actual Count
1	13 = 01101	13	169	$169/1170 = 0.14$	$169/293 = 0.57$	$\simeq 1$
2	24 = 11000	24	576	$576/1170 = 0.49$	$576/293 = 1.93$	$\simeq 2$
3	8 = 01000	8	64	$64/1170 = 0.05$	$64/293 = 0.21$	$\simeq 0$
4	19 = 10011	19	361	$361/1170 = 0.31$	$361/293 = 1.23$	$\simeq 1$
		Sum	1170			
		Avg.	293			

Page No.
Date: / /

STEP 5 Crossover & Mutation

Parent comb^n 1	1100 | 0	11001	→ 2
	0110 | 1	01100	→ 1
Parent comb^n 2	11 | 000	11011	→ 3
	10 | 011	10000	→ 4

STEP 6 Calculation of fittest offspring

String No	Offspring	x	$f(x) = x^2$
1	01100	12	144
2	11001	25	625
3	11011	27	729 ✓
4	1000	16	250

∴ Value has been maximised from 576 to 729.

Page No.
Date: / /

Artificial Neural Network

Machine learning (ML)

=> Branch of AI.

=> Uses mathematical formulas & statistical models to allow computer systems to perform tasks without explicitly programmed.

Types of ML

1) Supervised learning

=> Models are trained on labeled data i.e. input with known output.

=> $Y = f(x)$

output ← Y, x → input

=> Need supervision during training.

→ Data
- Train data → Train model → Test & validate model → Predit unlabelled data
- Test data → Test & validate model → Predit unlabelled data

Page No.
Date: / /

Applications :

→ Classification challenges eg. spam detection. image recognition.
→ Spam filtering
→ Fraud detection systems
→ Visual recognition systems

Disadvantages :

→ Difficult to classify big data
→ Requires labeled datasets
→ Needs a training process

2) Unsupervised learning

⇒ Models are trained on unlabeled data i.e. only input, no output.

⇒ Goal is to find hidden patterns or structures in the data.

⇒ Does not need supervision.

Unlabelled data → Training Model → Identify Patterns

Page No.
Date: / /

Applications :

→ Segmentation
→ Anomaly detection
→ Creating graphs, charts, images for decision making

Advantages :

→ Doesn't need labelled training data.
→ Can find previously unknown patterns.

Disadvantages

→ Often less accurate
→ Sensitive to data quality.

Supervised Vs Unsupervised Learning

	Supervised Learning	Unsupervised Learning
1	Learning using labeled data.	Learning using unlabeled data
2	More accurate	Less accurate
3	Can test the model easily with outputs.	Hard to test due to lack of outputs.
4	Simpler to implement.	Computationally more complex.

Page No.
Date: / /

Machine Learning

Supervised | Unsupervised

Classification | Regression | Clustering | Association

1) Classification

→ Assigning input data into predefined categories or classes.

Eg. Email spam detection, disease diagnosis.

2) Regression

→ Predicting a cts numeric value based on input data.

Eg. predicting house prices based on size, location etc.

3) Clustering

→ Grouping similar data points together based on patterns or features.

Eg. Customer prediction in marketing.

Page No.
Date: / /

4) Association

→ Finding rules or relationships bet^n variables in datasets.

Eg. Market analysis.

3) Reinforcement Learning

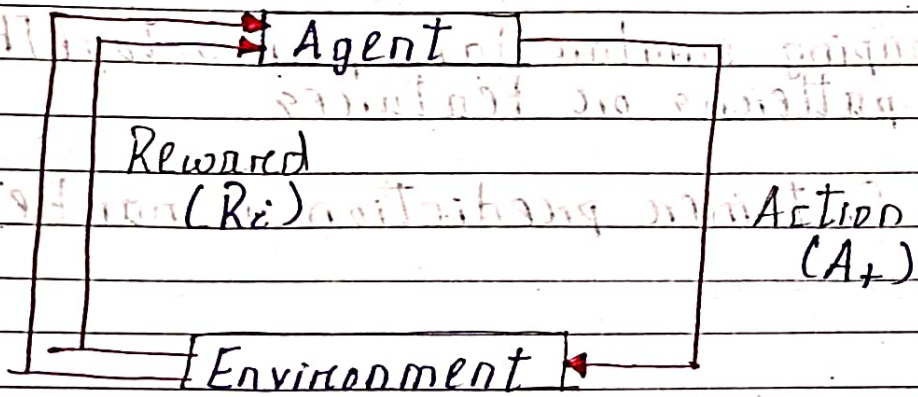
=> Agent interacts with the environment by sensing its state & taking actions.

=> The agent learns to take actions inorder to maximise long-term rewards.

=> It learns by trial & error, adjusting its behaviour to favour actions that lead to better outcomes.

=> This involves balancing rewards & punishments.

Eg. Beta version of windows.



Page No.
Date: / /

knowledge Based Agent

=> An intelligent agent needs knowledge about the world to make decisions & reason efficiently.

=> These agents can maintain an internal knowledge state & can update their knowledge based on observations & take actions accordingly

=> Components : 1) knowledge based
2) Inference Engine

Types of knowledge Representation

1) logical Representation
2) Structural knowledge
3) Production Rules
4) Frame Representation

1) Logical Representation

It is a method of representing knowledge using symbols & rules to describe facts & relationships respectively.

It is of 2 types :
i) Propositional Logic
ii) First order Logic

1. Write an example of each for strong AI and weak AI
2. Write two disadvantages of AI based applications
3. Give an example of AI based applications where it work like human

Page No.
Date: / /

1) Propositional logic

=> Simplest logic form where statements are represented as propositions.

=> Each proposition is a declared statement & logic is based on these.

Propositional Logic

- Atomic (T/F)
- Compound (Comb^n of 2 or more atomic proposition) ¬, ∧, ∨, →, <=>

Disadvantages

→ Cannot state relationships bet^n objects.
→ Only deals with declarative statements.

2) First Order Logic

=> Extension of propositional logic.

=> Define relationship bet^n objects, general rules & quantified statements.

Page No.
Date: / /

Basic Components of FOL

1) Const.s (1, 5, INDIA)
2) Variable (x, y, z, a, b)
3) Predicate
4) Func^n.s (sqrt, add (x, y))
5) Quantifiers ($\forall$ & $\exists$)
6) Connectives ($\wedge$, $\vee$, $\neg$, $\rightarrow$, $\leftrightarrow$)

1) Everyone at KIIT is smart.

$\forall x \{ KIIT(x) \rightarrow smart(x) \}$

2) All Indians are honest

$\forall x$ Indian $(x) \rightarrow$ honest (x)

3) There is a person who loves everyone in the world.

$\exists x \forall y$ loves (x, y)

4) Ram loves ice-cream.

loves (Ram, ice-cream)

5) All Roman was either loyal to Caesar or hate him.

$\forall x$ [Roman $(x) \rightarrow$ loyal $(x,$ Caesar$) \vee$ hate $(x,$ Caesar$)$

1. Write an example of each for strong AI and weak AI
of AI based applications
work like human

Page No.
Date: / /

6) Few person likes roses

$\exists x$ Person $(x) \wedge$ like $(x, Rose)$

7) Some physical objects are houses.

$\exists x$ house $(x) \wedge$ physical object (x)

8) Every house is owned by somebody

$\forall x \exists y$ house $(x) \rightarrow$ own (y, x)

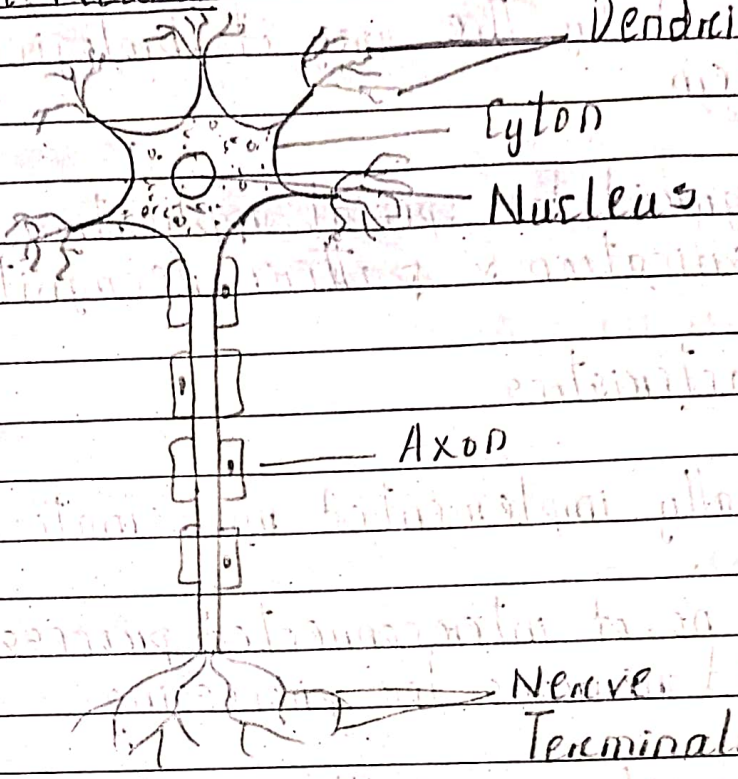
PL Vs FOL

	PL	FOL
1	Indivisible statements.	Predicate with arguments.
2	Cannot express relationships bet^n objects.	Can express object properties & relationships.
3	Simple reasoning with facts.	Complex reasoning in real world domain
4	Poor scalability	Better scalability

Page No.
Date: / /

ANN & Its Types

Biological Neuron



⇒ Human brain is composed of nerve cells called neurons & they are connected to other Thousand cells by axons.

⇒ Stimulus from external environment or I/P from sensory organs are accepted by dendrites.

Characteristics : i) Flexibility
ii) Robustness
iii) Parallel computation

2. Write two disad... of AI based ...

Page No.:
Date: / /

ANN

⇒ It is an info^n processing paradigm that is inspired by the way of biological nervous system.

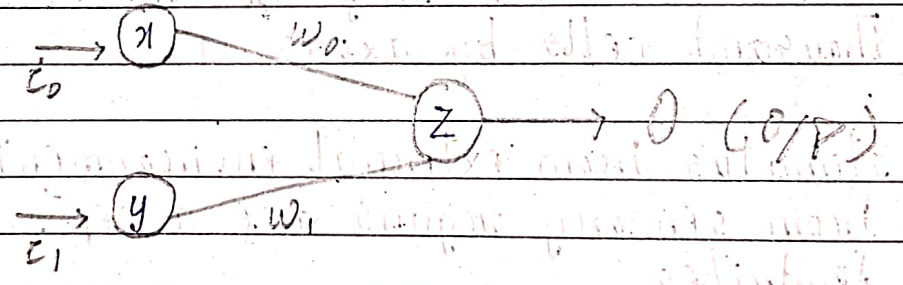
⇒ Configured for special application i.e. data classification & pattern recognition.

Characteristics

→ Neurally implemented mathematical model.

→ Huge no. of interconnected processing elements called neurons for processing.

→ I/P signals arrive at processing elements through connection & connected weights.

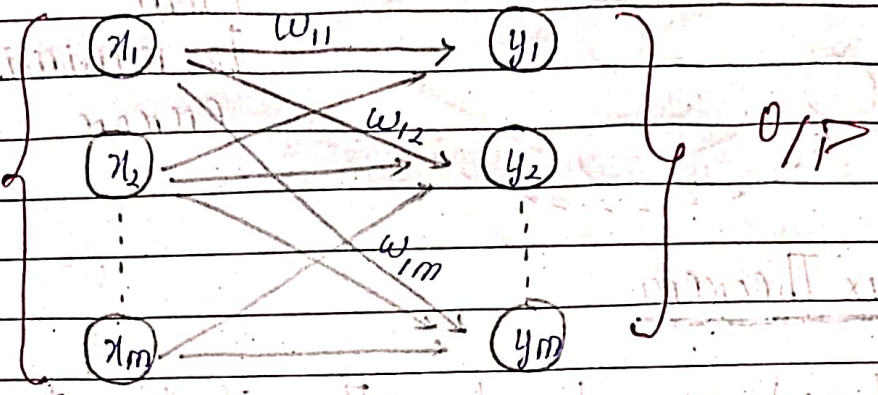


Input = $i_0 w_0 + i_1 w_1$ (Weighted sum)
Output = $f(I)$
↳ activation func^n

Page No.
Date: / /

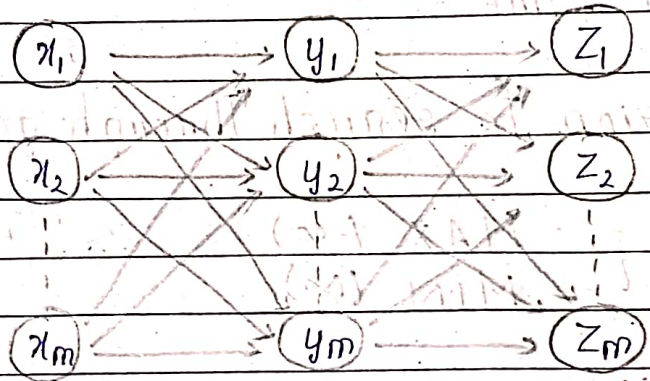
Types

1) Single layer feed forward N/W



2) Multilayer feed forward N/W

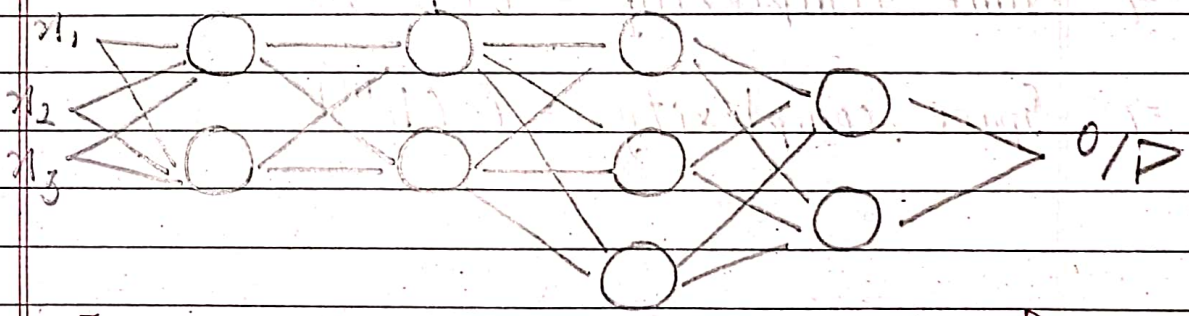
Computationally more stronger



↳ Hidden layer

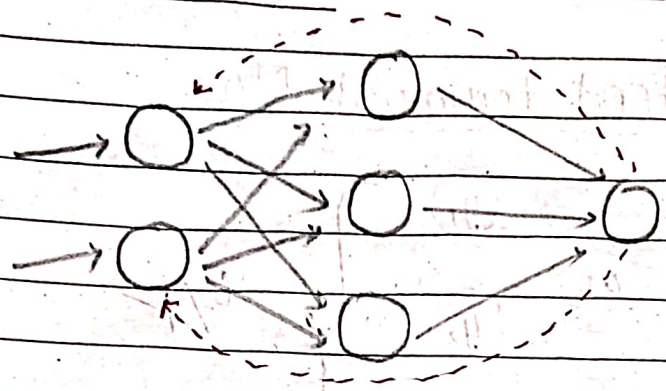
3) Multilayer Perception

Fully connected



3 or more layers are used to classify non-linearly separable data

4) Feedback ANN



Min Max Theorem

=> Specialized search algorithm that returns optimal sequence of moves for a player in zero sum game.

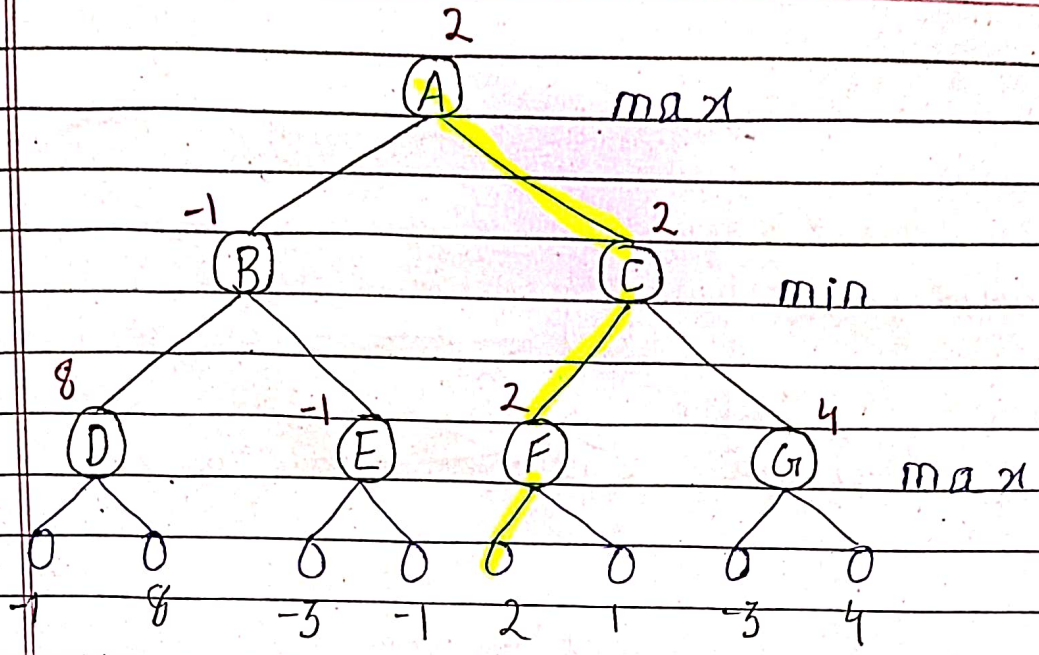
=> Uses recursion to search through game tree.

=> 2 players — MAX ($-\infty$)
MIN (∞)

=> DFS is used for exploration of complete game tree.

=> Time complexity = $O(b^m)$

=> Space complexity = $O(b^m)$


2
A
max
-1
2
B
C
min
8
-1
2
4
D
E
F
G
max
-1
8
-3
-1
2
1
-3
4