Name: Neeraj Rijhwani
Class: D15C
Rollno: 64

04/05

Assignment - 1 (AIDS)

1. What is AI? Consider the COVID-19 Pandemic situation, how AI helped to survive and renovated our way of life with different applications?

→ Artificial Intelligence (AI) refers to computer systems that can simulate human intelligence through learning, reasoning and problem-solving.

AI applications during Covid-19

(i) Disease Prediction and Diagnosis: AI helped predict outbreaks and diagnose COVID-19 using x-ray and CT scan analysis.

(ii) Chatbots & virtual Assistants: AI-Powered bots provide healthcare guidance and answered COVID-related questions.

(iii) Drug Discovery: AI analyzed potential drugs and vaccines, speeding up the research process.

(iv) Facial Recognition & Contactless Tech: AI helped in mask detection and automated temperature screening.

(x) Remote work & education: AI tools like zoom and google meet improved online collaboration

2. What are AI agents terminology, explain with examples

→ An AI agent is an entity that perceives its environment through sensors and acts upon it using actuators.

Types of AI agents:

- Simple Reflex agents: React to current condition
e.g thermostat.

- Model-Based Agents: Use internal models to make decisions (e.g self-driving cars).

- Goal-Based Agents: Take actions to achieve a goal.
e.g Chess-playing AI

- Utility-Based Agents: optimize performance based on preferences. (e.g recommendation systems)

- Learning Agents - Improve behavior using past experiences (e.g spam filters)

3. How AI technique is used to solve 8 puzzle problem?

→ The 8-Puzzle problem consider of a 3x3 grid with numbered tiles and one empty spac. The goal is to arrange the tiles in order.

AI techniques used:

(i) Breadth-first Search (BFS): Explores all possible possible moves level-wise.

(ii) Depth-first Search (DFS): Explores one path deeply before backtracking.

(iii) A* algorithm: Uses heuristics like Manhattan distance for optimal solution.

(iv) Greedly Search: focuses on moves that seems closer to the solution.

4. What is PEAS descriptor? Give PEAS description for following:

Taxi Driver
Medical diagnosis system
A music composer
An aircraft autolander
An essay evaluator
A robotic sentry gun for the Keck lab

System	Performance Marketing	Environment	Actuators	
Driver	Safety, time, efficiency	Roads, traffic	Steering, brakes	[illegible], Camer
Medical Diagnosis	Accuracy, treatment success	Patient health data	Alerts, treatment plans	Symp lab e
Music composer	Quality of music	Music database	Sound output	User in melody
→ Aircraft Autolander	Smooth landing	Weather, runway	Brakes, flaps	GPS, altim
→ Essay Evaluator	Grammar, clarity	Student essays	Feedback	Text a
→ Robotic Sentry Gun	Target accuracy	Security area	firing system	Motion, sensors camera

5. Categorize a shopping bot for an offline bookstore according to each of the 6 dimensions

→

Dimensions	Classification
Observability	Partially observable
Determinism	Stochastic
Episodic / Sequential	Sequential
Static / Dynamic	Dynamic
Discrete / Continuous	Discrete
Single / Multi-agent	Multi-agent.

6. Differentiate b/w model based and utility based agent.

→

Model Based	Utility Based
(i) Uses an internal model of the world	Chooses actions based on utility function.
(ii) Relies on past states and knowledge	Scales the best action based on utility.
(iii) Maintains an internal model of environment	Evaluates different possible outcomes and ranks them
Eg: Self-driving cars, robotics navigation	Movies recommendation systems, stock market trading AI.

7. Explain the architecture of a knowledge based agent and learning agent.

→ Knowledge Based Agent (KBA)

It is an AI system that stores knowledge, processes it and make decisions based on that knowledge.

Architecture.

- A KBA consists of following components:

(i) Knowledge Base (KB) : Stores fact and ...

(ii) Inference Engine : Uses logical reasoning to derive new facts.

(iii) Perception (Sensors) : Gathers information from environment.

(iv) Action Execution (Actuators) : Takes action based on reasoning.

Learning Agent.

A learning agent improves its performance over time by learning from past experiences.

Architecture:

(i) Learning Element : Learns from experience (e.g. neural networks)

(ii) Performance Element : Make decisions and takes action

(iii) Critic : Evaluates the agent's performance.

(iv) Problem Generator : Suggests exploratory actions for learning.

8. ~~tak~~ Same as Q1.

9. Convert the following Predicates:

a. Anita travels by car if available otherwise travels by bus.

→ Predicate: Travels (Anita, Car) ← Available (Car)

Predicate: Travels (Anita, Bus) ← ¬ Available (Car)

b. Bus goes via Andheri and Goregaon.

→ Predicate: Goes (Bus, Andheri) ∧ Goes (Bus, Goregaon)

c. Car has puncture, so it is not available

→ Predicate: ¬ Available (Car) ← Puncture (Car)

Given: Puncture (Car) = True, so Available (Car) = false.

Step-by step ~~forward~~ Reasoning:

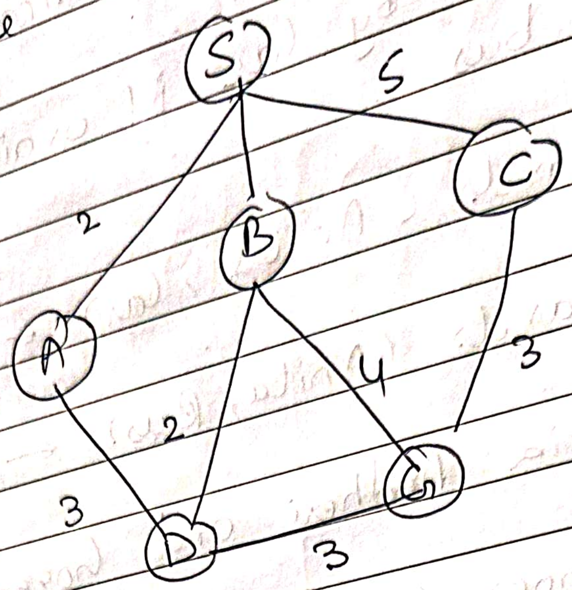
Step 1: Since Puncture (Car) = True, we get Available (Car) = false.

Step 2: Since Available (Car) = false, Anita must travel by Bus (Travels (Anita, Bus)).

Step 3: The Bus goes via Andheri & Goregaon (Goes (Bus, Goregaon))

Step 4: Since Anita is on the bus goes via Goregaon, Anita will travel via Goregaon.

10. Find the rout



→ BFS algorithm execution.

(i) Starts at S → add S to the queue [S].

(ii) Expand S → Add its neigbours [A, B]

(iii) Expand A → Add its neighbors (excluding S):
[B, C, D]

(iv) Expand B → Add its neighbors (excluding S)
[C, D, E]

(v) Expand C → No new nodes. Queue remains
[D, E]

(vi) Expand D → found G!

Shortest Path by BFS.

S → A → D → G

11. Explain depth limited search? Explain Iterative Deepening Search with example.

→ Depth-limited Search (DLS)

It is a variation of Depth-first Search (DFS) where a depth limit is set to avoid infinite recursion in large or infinite search spaces

How it works?

- Explores nodes depth-wise like DFS but stops at a specific depth.

- If the goal is not found, it does not continue searching deeper.

Advantages:
- Avoids infinite loops in large graphs.
- Uses less memory than BFS.

Disadvantages:
- If goal node is beyond the depth limit, it won't be found.

- Choosing the right depth limit is tricky.

Eg. If the Depth limit is set to 2, DLS will only explore nodes within 2 levels of depth, ignoring deeper paths

Iterative Deepening Sea...

It combines BFS and DLS gradually the depth limit set by step until the goal is found.

How it works?
- Performs DLS with depth = 0
- Increases depth limit to 1, then 2, then 3 and so on.
- Stop when goal is found.

Advantages:
- finds the shortest path like BFS
- Uses less memory than BFS

Disadvantages:
- Some nodes get revisited multiple times making IDS ~~sligkrightly~~ slightly slower.

Eg: If the goal is at depth 2, IDS will explore:

Depth 0: Start Node
Depth 1: level 1 nodes
Depth 2: level 2 nodes
Depth 3: level 3 nodes
Depth 4: Goal found

12. Explain how climbing and its drawbacks

in detail with example. Also state limitations of Steepest - ascent hill climbing

→ Hill Climbing is an AI algorithm that continuously moves towards the best immediate solution to maximize / minimize an objective function.

How it works?

(i) Start with an initial state
(ii) Evaluate all possible next states.
(iii) Move to the best neighbouring state (higher value).
(iv) Repeat until no better moves exist.

Eg: Imagine climbing a mountain in fog where you can only see a few steps ahead. You always move uphill but might get stuck at a local peak.

Drawbacks of Hill Climbing

(i) Local Maxima: The algorithm might stop at a peak that is not the best solution.

(ii) Plateau Problem: The algorithm get stuck in flat areas where no uphill moves exist.

(iii) Ridges: It cannot make lateral moves to find better paths.

Example: In a TSP (travel... hill climbing might find a sub... optimal if the algorithm get stuck ... ~~maximum~~ minimum.

Limitations of Steepest-Ascent Hill Climbing

(i) It chooses the best possible move at each step but does not consider future paths.

(ii) Risk: Can miss better solutions if they req a temporary bad move to reach a higher peak.

(iii) Solution: Use Simulated Annealing or Random Restart Hill Climbing

13. Explain simulated annealing and write its algorithm.

→ Simulated Annealing (SA) is an AI optimization technique that allows occasional bad moves to escape local maxima and find a better global solution.

How it works?

(i) Start with an initial solution
(ii) Evaluate neighboring solutions

FOR EDUCATIONAL USE

...ution is better, accept it.
If the new solution is worse, accept it with some probability (controlled by a decreasing "temperature" factor)

(v) Gradually reduce temperature, lowering the chances of accepting bad moves.

(vi) Stop when temperature reaches zero or no further improvements can be made.

14. Explain A* algorithm with an example.

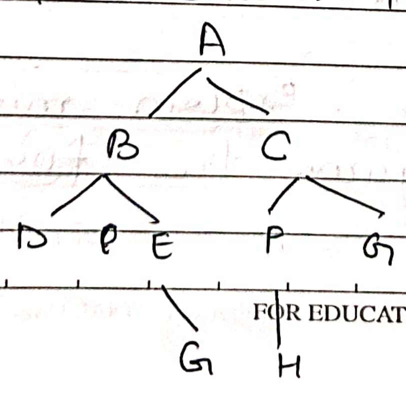
→ A* algorithm is a graph search algorithm that finds the shortest path efficiently using:

g(n): Cost to reach the current node.

h(n): Estimated cost from the current node to the goal (heuristic)

$f(n) = g(n) + h(n)$ (total estimated cost)

Example: find shortest path from A to H



Assume costs:

g(n) : Distance from start to the goal

h(n) : Estimated distance

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

Node	g(n)	h(n)	f(n)
A	0	6	6
B	2	4	6
C	3	3	6
D	4	5	9
E	5	3	8
F	6	2	8
G	7	0	7

Algorithm execution:

(i) Start at A ($f(A) = 6$)

(ii) Expand B & C ($f(B) = 6$, $f(C) = 6$)

(iii) Expand C (lower heuristic)

(iv) Expand F → then G (goal reached)

final Path : A → C → F → G

15. Explain Min Max. Explain minimax algorithm and draw game tree for Tic TAC Toe Game.

→ Minimax Algorithm is used in two-player turn-based games (like Tic Tac Toe, Chess). It helps a player choose the best move by minimizing the opponent's best possible advant--age.

Key concepts:

(i) MAX Player: Tries to maximize the score (e.g AI in Tic Tac Toe)

(ii) MIN Player: Tries to minimize the score (e.g Human opponent)

(iii) Game Tree: A tree showing all possible moves and their outcomes.

(iv) Leaf Nodes: Represent the game's final outcome (Win = +1, Lose = -1, Draw = 0).

Tic Tac Toe Game Tree Example.

Consider a Tic Tac Toe board where it's MAX's turn (X) and the board looks like this,

X	O	X
O	X	O
-	-	-

Possible moves for X:

1. Move in (2,1)
2. Move in (2,2)

Minmax Tree Representation

(X's Turn)

-1 (O's move) 0 (O's move)

Here, max will choose 0 because -1 is worse.
Final move : X moves to (2,2) to get the best possible outcome.

16. Explain Alpha beta pruning algorithms for adversial search with example.

→ It is an optimization technique for

...ed ... it eliminates branches that don't
time. to be evaluated, reducing computa

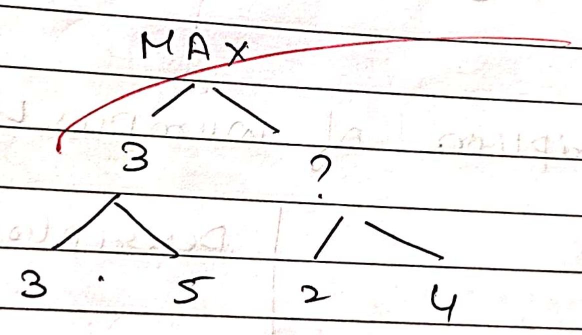
- Why Use Alpha-Bet Pruning?

(i) Speeds up the Minmax Algorithm

(ii) Ignores unnecessary calculations, leading to faster AI decision-making

Alpha-Beta Pruning Example

Consider a game tree where the MAX player wants to maximize the value, and the MIN player wants to minimize:



→ without pruning: The AI checks all nodes

→ With pruning: If a node's value is already worse than a known value, we ignore it.

17. Explain WUMPUS world environment giving its PEAS description. Explain how percept sequence is generated?

→ The wumpus world is a grid-based environment used to test AI logical reasoning. It consists of:

Agent → AI that explores the world.

Wumpus (Monster) → Kills the agent if encountered.

Pits → Traps that kills the agent.

Gold → The agent must collect gold and exit safely

PEAS description of wumpus world

	Component	Description
(i)	Performance Marketing	+1000 for gold, -1000 for death, -1 per move.
(ii)	Environment	4x4 grid with wumpus, Pits and Gold.
(iii)	Actuators	Move (left, right, up, down) Grab, shoot.
(iv)	Sensors	Smell (near wumpus), Breeze (near pit), Glitter (near Gold)

...owing crypto arithmetic problems
SEND + MORE = MONEY

→
$$
\begin{array}{ccccc}
 & S & E & N & D \\
+ & M & O & R & E \\
\hline
M & O & N & E & Y
\end{array}
$$

(i) Assign unique digits

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

(ii) ~~9 5~~ Verify the sum:

$$
\begin{array}{ccccc}
 & 9 & 5 & 6 & 7 \\
+ & 1 & 0 & 8 & 5 \\
\hline
1 & 0 & 6 & 5 & 2
\end{array}
$$

19. Consider the following axioms:

→ Given axioms

(i) All people who are graduating are happy

(ii) All happy people are smiling

(iii) Someone is graduating

(i) Step 1: Convert to First Order Predicate Logic (FOL)

Graduating $(x) \rightarrow$ Happy (x) $\rightarrow$ (A1) $\rightarrow$ People who graduate are happy

Happy (x) $\rightarrow$ Smiling (x) $\rightarrow$ (A2) $\rightarrow$ Happy People are smiling

$\exists x$ Graduating (x) $\rightarrow$ (A3) $\rightarrow$ Someone is graduating

(ii) Step 2: Convert to Clause Form

- Convert implications $(\rightarrow)$ to Clauses $(\neg P \vee Q)$

Graduating (x) $\rightarrow$ Happy (x) $\rightarrow$ $\neg$ Graduating (x) $\vee$ Happy (x)

Happy (x) $\rightarrow$ Smiling (x) $\rightarrow$ $\neg$ Happy (x) $\vee$ Smiling (x)

- Convert Existential Quantifier $(\exists)$ to Skolem Constant (e.g., let a be a person who is graduating)

$\exists x$ Graduating (x) $\rightarrow$ Graduating (a)

Now the clause form is;

...duating (x) ∨ Happy (x)

¬ Happy (x) ∨ Smiling (x)

Graduating (a) (From ∃x Graduating (x), we replace x with constant a)

(iii) Step 3: Prove "Someone is smiling" Using Resolution

We need to prove Smiling (a) by applying resolution step-by-step.

- Unify Graduating (a) with ¬ Graduating (x) ∨ Happy (x):

Graduating (a)
¬ Graduating (a) ∨ Happy (a)

Happy (a)

- Unify Happy (a) with ¬ Happy (x) ∨ Smiling (x):

Happy (a)
¬ Happy (a) ∨ Smiling (a)

Smiling (a)

FOR EDUCATIONAL USE

(iii) Resolution Tree

¬Happy (a) ∨ Smiling (a)

↑

Happy (a)

↑

Graduating (a)   ¬Graduating (a) ∨ Happy (a)

↑

Given Fact

20. Explain Modus Ponen with suitable example.

→ Modus Poneus (MP) is a logical inference rule that states:

(i) If P → Q (i.e, "If P is true, then Q is true")

(ii) P is true

(iii) Therefore, Q must also be true

Example of Modus Ponens

(i) If it rains, the ground is wet → Raining (x) → Wet Ground (x)

(ii) It is raining → Raining (Today)

(iii) Therefore, the ground is wet → Wet Ground (Today)

P   q (Premise 1)
(Premise 2)
∴ Q (Conclusion)

Real - world Example:

(i) Premise 1: If a student studies hard, they will pass the exam (Study (x) → Pass (x))
(ii) Premise 2: The student studies hard (Study (Neeraj))
(iii) Conclusion: The student will pass the exam (Pass (Neeraj))

Q1. Explain Forward ~~chaining~~ chaining and backward chaining algorithm with the help of example

→ Forward Chaining

- It is a data-driven approach that starts from known facts and derives conclusion step by step.

Steps:

(i) Start with initial facts
(ii) Apply Inference rules to generate new facts

(iii) Continue until the goal is reached.

Example: Use Cases

- Expert systems, rule-based AI (e.g. medical diagnosis)

Backward chaining

It is a goal-driven approach that starts from the goal and works backward to determine if fact support it.

Steps:

(i) Start with goal (G)
(ii) Check if G is directly known in knowledge Base (KB)
(iii) If not, check if G can be derived from other rules
(iv) Repeat until known facts are reached.

Example: Use Cases:

- AI planning, medical diagnosis, legal reasoning.