Nikita Thadani
D15C
55

AIDS Assignment 1

A+1
8½/10

Q1] What is AI? AI's Role in COVID-19 Pandemic

→ AI in machines refers to the simulation of human intelligence in machines. AI systems can perform tasks such as learning, reasoning, problem solving, perception and language understanding. AI is categorized in various types also.

AI played a crucial role in various aspects of healthcare & dailylife, such as:

Early detection & diagnosis: AI-based models detected & analyzed patterns

Medical research & drug discovery: It discovers potential treatments

Remotetec work & learning: Online learning & remote collabration

Q2] What are AI agents terminology, explain with example

→ An AI agent is an entity that provides perceives its environment through sensors.

Agent Function :- The agent function of an agent specifies the action taken by the agent in response to any percept

Performance Measure :- It evaluates the behaviour of the agent in an environment

Rational Agent :- A rational agent acts so as to maximize the expected value of performance measure.

Task Environment :- It includes the performance measure, the external environment, sensor & acutators.

Q3] How AI agents are used to solve 8 puzzle problem?

→ The 8 puzzle problem is a sliding puzzle consisting of 3×3 grid with 8 tiles & one empty space. Techniques to solve are:

- Breadth-First-Search: Explores all possible moves level by level
- Depth First Search- Explores all deeper path First
- A* Algorithim: Uses a heuristic function to find optimal path

Q4] Describe PEAS Framework

→

| AI system | Performance | Environment | Actuator | sensor |
|---|---|---|---|---|
| • Taxi Driver | Safety, speed | Roads, traffic | Steering, breaking | GPS, cameras |
| • Medical Diagonsis | Accuracy, treatment success | Symptoms, Diseases | Prescription | Test, patient history |
| • Music composer AI | Harmony | Instruments, Music notes | Speaker | Genre data, User preference |
| • Aircraft Autolander | Landing precision | Runway. wind, altitude | Flaps, engines | GPS, gyroscope |

| | | | | |
|---|---|---|---|---|
| Essay Evaluator | Accuracy, grammar check | Essay | Score system | Text reconition |
| Robotic Sentry Gun | Threat detection, accuracy | security area | Gun movement | Camera Motion sensor |

Q5] Categorize a shopping bot for an ffonline bookstore according to 6 dimensions

→ Observable : Partially observable
Determinism / Stoctiastic : Deterministic
Episodic / Sequential : Sequential
static / Dynamic : Dynamic
Discrete / continous : Discrete
Single / Multi agent : Single Agent

Q6] Difference between Model and utility based agent

| Feature | Model Based | Utility Base |
|---|---|---|
| Internal Model | Maintains an internal model of world | Evaluates actions based on utility |
| Decision Making | Uses Past state | Selects actions to maximize utility |
| Example | Self driving car | AI recommdation |

Q7] Architecture of Knowledge Based & Learning Agents.

→ Knowledge Based Agent:
Uses a knowledge base to store information
Employs interfrence rules to make decision
Example: Medical diagnosis

Learning Agent:
Improves it performance using feedback loops
Composed of learning elements, critic, performance element & problem Generator
Example: AI in gaming

Q8.] Predicate Logic conversion for the following.

→ a) Anita travels by car if available, otherwise travels by bus
$\forall x$ (Car Available (x) → Travels (Anita, Car))
$\exists$ (Car Avaible (x) → Travels (Anita, Bus)

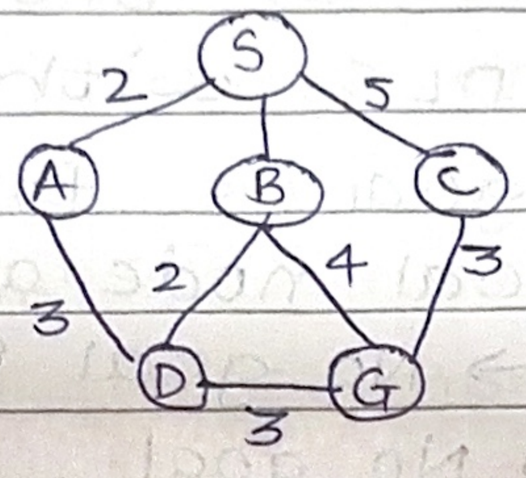
b) Bus Goes via Andheri & Goregaon
Goes(Bus, Andheri) ∧ Goes (Bus, Goregaon)

c) Car has a puncture, so it is not available
Has Puncture (car) → $\exists$ Car available (car)

Solution: Yes, Anita will travel via Goregaon

Page No.
Date

Q9] Using BFS, travel from S to G




S → A (cost 2)
S → B (cost 5)
S → C (cost 5)
A → D (cost 3)
A → G (cost 3)
C → G (cost 4)

Queue,
1) Queue [S]
2) S → Queue [A, B, C]
3) A → Queue [B, C, D, G]
4) B → Queue [C, D, G]
5) C → Queue [D, G, G]
6) D → Queue [G, G]
7) G → Goal found

Shortest path is : S → A → G
Total cost : 2+3=5

Q10] Explain Depth limited search & IDS

→ A variant of DFS that limits the depth of recursion to a fixed value L
Avoids infinite loops in deep or infinite state spaces.
Limitations: If L is too small, it may fail to find a solution
Ex: If a goal node exists at depth 5 & L=3, DLS wont find solution.

Iterative Deepening search (IDS):
Combines the benefits of BFS & DFS
Repeatedly applies DLS with increasing
depth limits until goal is found.
Example: Find a goal node at depth 3
1) Apply DLS with L=0 → No goal found
2) Apply DLS with L=1 → No goal
3) Apply DLS with L=2 → No goal
4) Apply DLS with L=3 → Goal found

Q11] Explain Hill Climbing
→ Hill climbing is a heuristic search algo that moves towards the most promising direction to find optimal solution
Types:
1) Steepest-Ascent hill climbing
2) Simple hill climbing
3) Stochastic hill climbing

Drawbacks:
Local maxima: stuck in a suboptimal peak
Plateau: No clear direction to move
Ridge Problem: Moves may not directly increase
Ex: Navigating a mountain to reach the highest peak

Limitations of steepest-Ascent hill:
- Can get stuck at local optima
- No backtracking
- Does not work well for complex spaces

Page No.
Date

Q12.] Simulated annealing is a probabilistic
→ search technique that avoids getting trapped in local optima by allowing occasional bad moves.

Algorithim:

Initialize temperature T, cooling rate k
Choose an intial solution S
Repeat unit T→0:
a. Select a neighbouring solution s'
b. Compute cost diff. $\Delta E = Cost(s') - Cost(s)$
c. If $\Delta E < 0$, move to s'
d. Else, move to s' with prob $e^{(-\Delta E/T)}$
e. Reduce T: $T = \alpha * T$
Return best solution found

Q13.] ~~A* is~~ Explain A* algo with example
→ A* is an informed search algorithm tha uses:
$f(n) = g(n) + h(n)$
where $g(n)$ = cost from start to node n
$h(n)$ = heuristic
Ex: Finding shortest path in a graph like google maps
It is widely used in pathfinding, AI & game
Uses both $g(n)$ & $h(n)$
A* is complete
A* is optimal

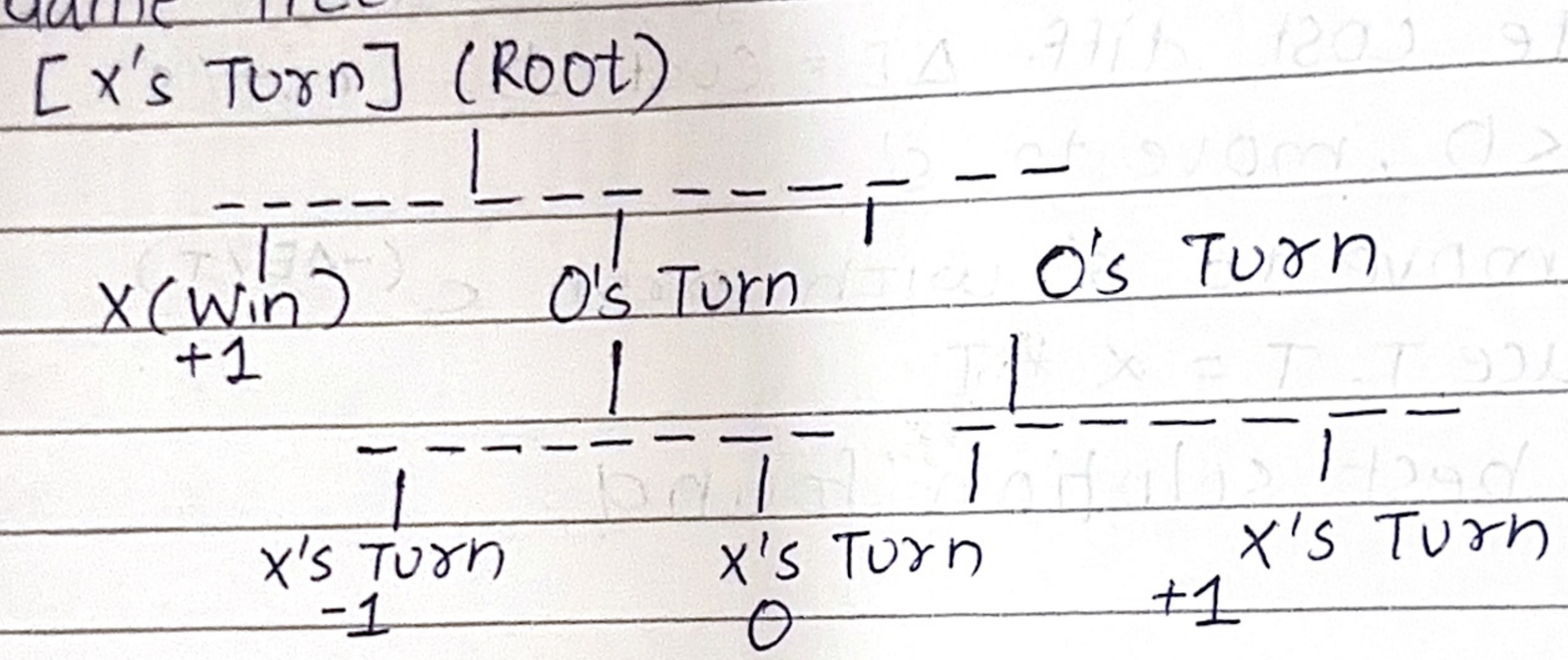
Q14] Give MinMax algorithm

→ (Tic-Tac-Toe Game Tree)

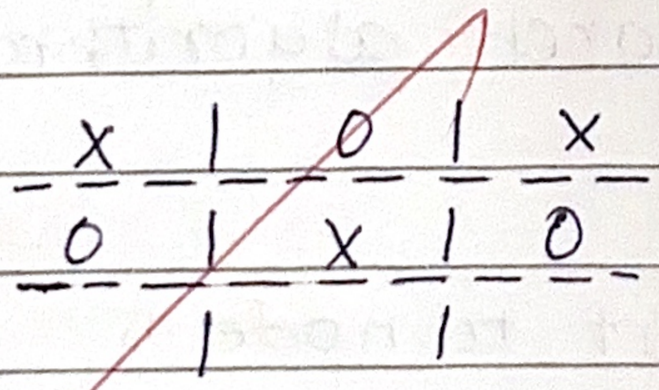
MinMax is an adversial search algorithm used for two-player games (like chess, tic-tac-toe)

- Min-player: Tries to minimize opponents sc
- Max-player: Tries to maximize their score

Game Tree:




Current state of game

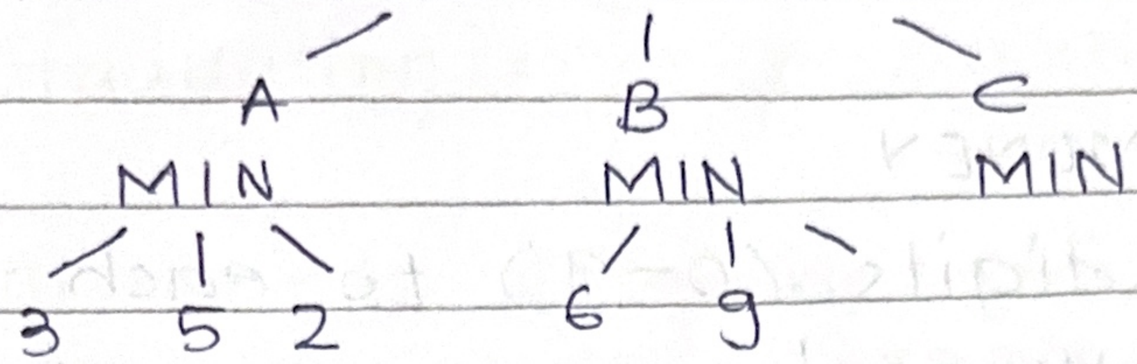



Q15] Explain Alpha Beta Pruning algo

→ Algorithm -

1. Start at the Root ($\alpha = -\infty, \beta = +\infty$)
2. Evaluate Node A & its child values Seled the lowest value & update $\beta$
3. Move to Node B evaluate its first child
4. Move to Node C

Eg:




In this-
- Start at Root
- Evaluate Node A & it child for lowest value i.e 2
- Move to Node B, here first child is 6
  ∴ 6>2, we prune this subtree
- Move to Node C & evaluate
  ∴ 6>2, max will not pick this
- Max picks Node A with value 2

Q17. Explain Wumpus world environment

→ The Wumpus world is a simple grid-based environment used to demonstrate logical reasoning in AI. It consists of:
A cave (4x4 grid) with hidden dangers
A wumpus (dangerous monster)
Pits (falling in them is death)
Gold (the player's gold)

PEAS :-
Performance - Penalty for death, reward
Environment - Wumpus, pits
Actuators - move, grap
Sensors - breeze, scream

Q18] Solve

1. SEND + MORE = MONEY

→ Assign unique digits (0-9) to each letter & no two letters have same value

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

Since MONEY has 5 digits M = 1

Assigning the maximum value to S = 9

Accordinly, we assign values & solve

S = 9 N = 6 M = 1 R = 8

E = 5 D = 7 O = 0 Y = 2

```
  9 5 6 7
+ 1 0 8 5
---------
1 0 6 5 2
```

Q19] Consider the anxioms and answer the questions.

→ 1. All people who are graduating are happy

$\forall x$ (Graduating (x) → Happy (x))

2. All happy people are smiling

$\forall x$ (Happy (x) → Smiling (x))

3. Someone is graduating

$\exists x$ (Graduating (x))

Converting :-

1. Graduating (x) → Happy (x) → ¬Graduating (x) ∨ Happy (x)

2. Happy(x) → Smiling(x) → ¬ Happy(x) ∨ Smiling(x)
3. ∃x(Graduating(x)) → Graduating(a)

To prove - Is someone smiling
1. ¬ Graduating(a) ∨ Happy(a)
2. Happy(a) ∧ smiling(a)
3. Smiling(a)

Hence, proven

Tree:

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

Happy(a)  ¬Happy(a) ∨ Smiling(a)

Similing(a) - Proved

Q20] Explain Modus Ponen with example

Modus Ponens is a fundamental rule in logic in that states:
- If P → Q (IF P, then Q)
- And P is true,
- Then Q is true

Example:
1. If it rains, the ground will be wet (Rains → Wet Ground)
2. It is raining. (Rains = True)
3. Therefore, the ground is wet (Wet Ground = True)

i.e $\frac{P \rightarrow Q, P}{Q}$

Page No.
Date

Q21] ~~EO~~ Explain Forward Chaining & Backward Chaining.

→ Forward chaining: starts with known facts until reaching a conclusion.
It is used in expert systems

Example
1. Fact - It is cloudy
   Clouds cause rains
2. Rule : Cloud → Rain
3. Conclusion: It will rain

Backward chaining:
Starts with the goal and works backward to see if facts support it
Used in prolog & theorems

Example
1. Goal : Will it rain?
2. Rule : Cloud → Rain
3. Check: Is it cloudy
4. If yes,
   Conclusion: Rain = True