Analytical

1) Analyze the diversity of environment in which intelligent agents operate, ranging from deterministic to stochastic, observable to partially observable, and discrete to continuous. Discuss how the characteristic of environment influence the design and behavior of agents, including their sensing capability, action space, and decision-Making processes.

Evaluate the challenges posed by dynamic and uncertain environments and the strategies employ to adapt and succeed.

Answer:-

1) Types of Environment:-

* Deterministic / Stochastic
* observable / partially observable
* Discreate / Continuous
* Static / Dynamic

2) Influence on Agent Design & Behavior:-

* Sensing Capability:- Depends on observability, better Sensor for partial observability.

* Action Space:

Discrete actions require predefined sets ; Continuous requires Complex Control mechanisms.

* Decision - Making:

Deterministic & static : use simple rule-based or search algorithms

3) Challenges in Dynamic & Uncertain Environments:

* Unpredictable state changes due to randomness or other agents.

* Incomplete or noisy sensor data.

* High Computational Complexity in Continuous or large action spaces

4) Adaptation Strategies for Agents:

* probilistic reasoning (eg, Bayesian networks)

* Machine learning and reinforcement learning

* Sensor fusion to Combine multiple noisy inputs for better accuracy

* prediction and planning modules to anticipate future scenarios.

2) Analyze the relationship between the nature of the environment and the structure of an intelligent agent. How does the Complexity, observability, and dynamicity of the environment influence the design and functioning of agents? Discuss various agent architectures and their suitability for different types of environment.

1) Relationship Between Environment and Agent Structure:

* The environments nature (Complexity, observability, dynamicity) directly affects how an agent is designed and functions.

* Complex environment need agents with advanced reasoning and adaptability.

* Partially observable environment require strong sensing and inference abilities.

* Dynamic environments demand real-time perception and decision-Making Capabilities.

2) Influence of Environment Characteristics:

* Complexity :-

More Complex environments → agents need better memory, planning, and learning abilities.

* Observability: Fully observable → Simple reactive agents
Suffice.

Partially observable → require inference, belief states, or history-based decision Making.

* Dynamicity :-

static → Can use pre-Computed plans.

Dynamic → need Continuous Sensing and adaptive actions.

3) Develop a problem-solving agent for a robotic arm that ~~Solv~~ Sorts Colored

PEAS :- Performance measure, Environment / Actuators, Sensors
medical Diagonasis System.

Performance measure :-

Accuracy of diagonsis, Speed of diagonal patient
Safety minimal false positive / negative.

Environment :- Hospital patient, Health records, clinical data
medical data base.

4) Discuss the properties of environment. How does the vacuum cleaner perceive its environment? What sensing mechanisms are employed and their role in detecting dust, obstacles etc.

A robotic vacuum cleaner operates in a partially observable, dynamic, stochastic, continuous, and sequential environment.

Sensors used:-

* IR sensors : Detect obstacles
* diff sensor : Detect stairs / drops
* Bump Sensor : Detect collisions
* Dirt Sensor : Detect dust concentration
* Wheel encoders & IMU : Track position and motion
* LIDAR / camera (in advance models) : Map surroundings

Actuators Used:-

* Wheel Motors : Control movement (forward, backward, turns)
* Brush motors : clean surfaces
* Suction motor : Lift dirt

Decision Making process:-

1) Sense : Gather data from sensors
2) Analyze : Detect obstacles, dirt areas, cliffs and free paths.

3) Decide:

* If obstacle ahead : Stop, turn
* If cliff detected : Reverse and turn.
* If high dirt detected : Slow down, increase Suction, clean thoroughly
* If clear path : Move forward
* If at room edge : follow wall-cleaning pattern

4) Act : Move, turn, adjust cleaning modes as per decision.

Sensors Continuously Scan the environment

⇓

Controller analyzes inputs

⇓

Select actions like move forward, turn left, turn right, reverse, clean deeply, or stop

⇓

actuators execute those actions.

---

5) Explore the Search Space using strategies and formulate the problem Components for the 8-queens problem using the following information.

→ place 8 queens on a chessboard Such that none of the queen's attack any of the other. A Configuration of 8 queens on the board as Shown in the figure below, but this does not represent a solution as the queen in the first Column is on the Same diagonal as the queen in the last Column.

Problem:

place 8 queens on an 8x8 chessboard so that no two queens attack each other (no same row, column, or diagonal).

Problem Formulation Components:-

Components	Description
Initial State	– Empty chessboard with no queens placed
State Space	– All possible ways to place 1 to 8 queens on the board without conflict.
Action	– place a queen in a row such that it doesn't attack previously placed queens
Transition model	– Move from one state (board configuration) to another by placing a queen safely.
Goal Test	– 8 queens are placed on the board with none attacking each other.
Path Cost	– Typically uniform or zero (Since we're interested in reaching a valid configuration not minizing moves).

Search Space Exploration:

* No other queen in the same column
* No other queen is on the same major or minor diagonal

Total possible configuration (without constraints)

$$8^8 = 16,777,216$$

Search Strategies for 8-queens:-

Depth-first Search (DFS):-

place a queen in a row, move to next row, Backtrack if Conflict

Breath first Search:

Explore all possible placements row by row

Backtracking (most Common):-

DFS + Constraint checking: place queen only in Safe position:

Heuristic Search (eg: Hill climbing)

Use a heuristic (like number of attacking pairs) to guide placements.

Local Search:

Start with a full board and move queens to reduce Conflicts.

---

6) Solve the water Jug problem: You are given 2 jugs, a 4-gallon one and 3-gallon one. Neither has any measuring marker on it. There is a pump that can be used to fill the Jugs with water. How can you get exactly 2 gallons of water into 4-gallon jug? Explicit assumptions: A Jug can be filled from the pump, water can be poured out of a jug onto the ground, water can be poured from one jug to another and that there are no other measuring devices available.

We have:

1 x 4 - gallon jug

1 x 3 - gallon jug

Operations allowed:

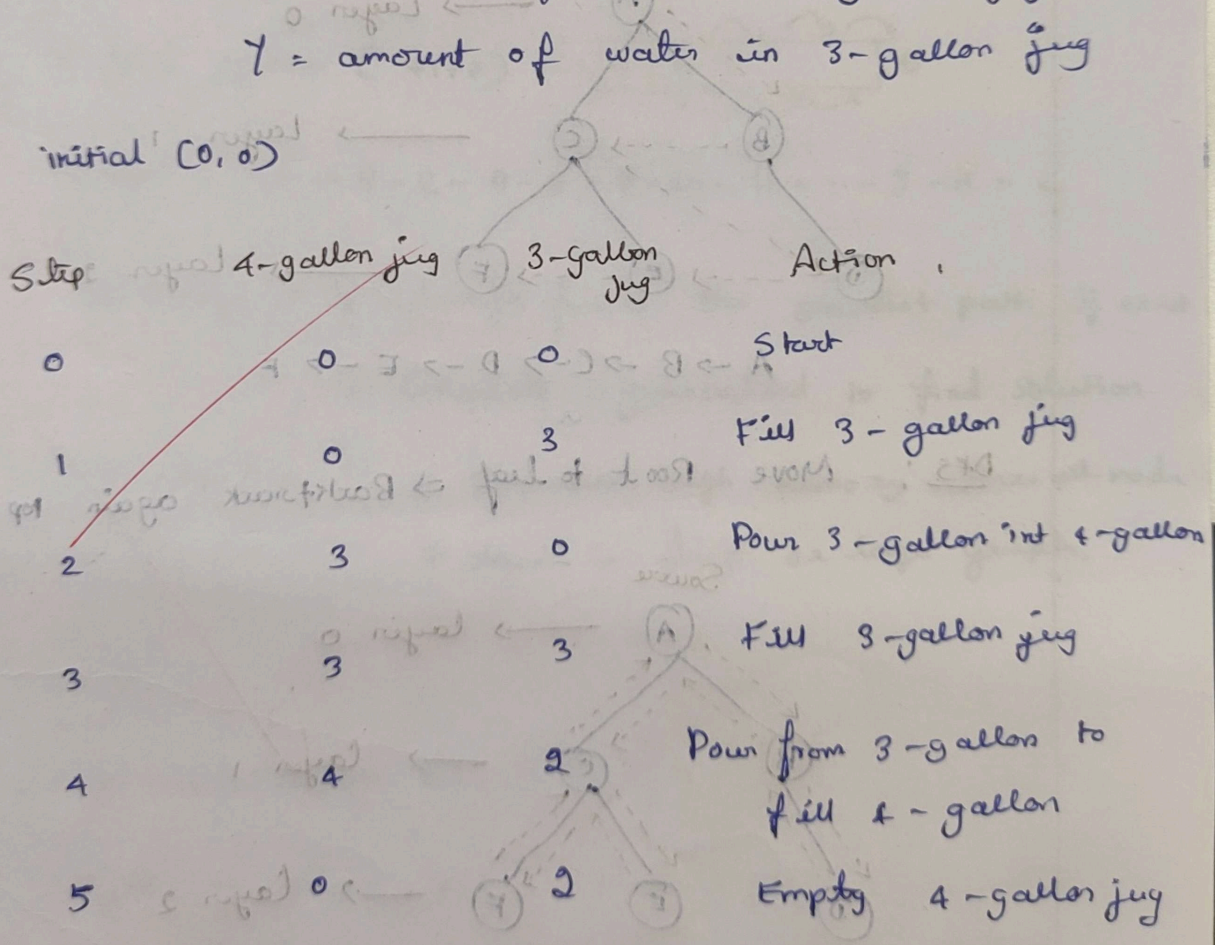
* Fill the jug Completely
* Empty any jug Completely
* pour water from one jug to the other until either the source jug is empty or the target jug is full.

Goal: Get exactly 2 gallons in the 4-gallon jug

Solution

x = amount of water in 4-gallon jug

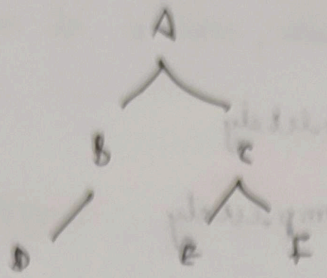
y = amount of water in 3-gallon jug



initial (0, 0)

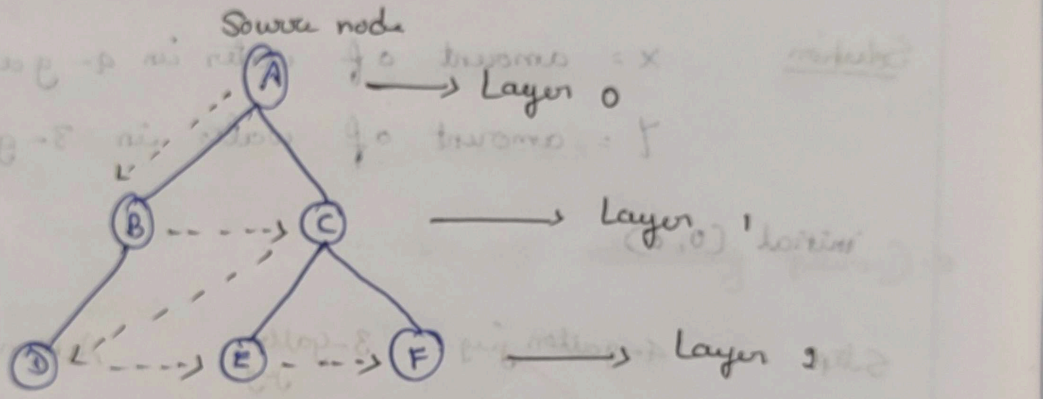
Step	4-gallon jug	3-gallon jug	Action
0	0	0	Start
1	0	3	Fill 3-gallon jug
2	3	0	Pour 3-gallon into 4-gallon
3	3	3	Fill 3-gallon jug
4	4	2	Pour from 3-gallon to fill 4-gallon
5	0	2	Empty 4-gallon jug
6	2	0	Pour remaining 2 gallons to 4-gallon

7) Show how BFS & DFS work on the Search tree for given graph.



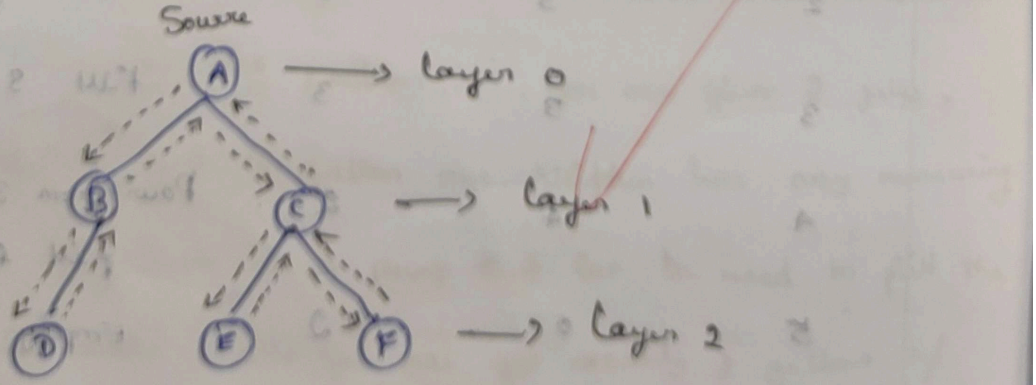
Solution:

BFS: move from left to right



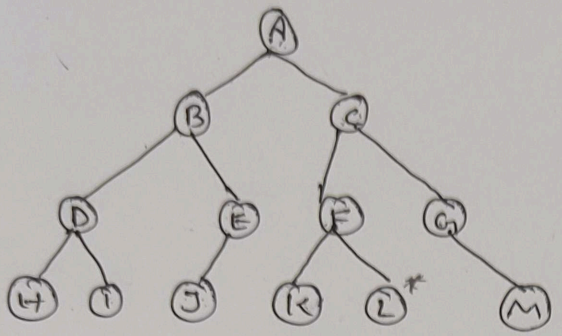
A -> B -> C -> D -> E -> F

DFS: Move Root to leaf => Backtrack again top to down



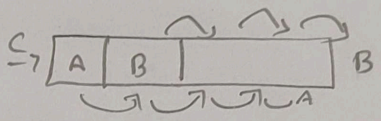
A -> B -> D -> C -> E - F

8) Discuss uninformed Searching Strategies BFS and DFS with its advantages and disadvantages using the following graph reach the goal "L".



Target = L

Uninformed BFS:-

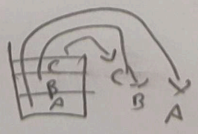
* Uses queue (FIFO) 

Order:- A → B - C - D - E - F - G - H - I - J - K - L

Advantage:- * Always finds the shortest path if exist
* Complete - guaranteed to find solution

Disadvantage:- * Consumes high memory - Store all node
* Slower in deep or large graphs.

Uninformed DFS:-

* uses stack (LIFO) 

Order:- A → B - D - H - I - E - J - C - F - K - L

Advantage: → Less memory usage
→ Find solution without exploring all node

Disadvantage: → Not guaranteed to find the shortest path
→ May enter in finite loop
→ incomplete