Artificial Intelligence

Unit 1

1.Define Artificial Intelligence, Intelligent Agent and its use.

a. **Artificial Intelligence** is composed of two words **Artificial** and **Intelligence**, where Artificial defines "*man-made,*" and intelligence defines "*thinking power*", hence AI means "*a man-made thinking power.*"

"It is a branch of computer science by which we can create intelligent machines which can behave like a human, think like humans, and able to make decisions."

Artificial Intelligence exists when a machine can have human based skills such as learning, reasoning, and solving problems.

With Artificial Intelligence you do not need to preprogram a machine to do some work, despite that you can create a machine with programmed algorithms which can work with own intelligence, and that is the awesomeness of AI.

b. **An intelligent agent** is an autonomous entity which act upon an environment using sensors and actuators for achieving goals. An intelligent agent may learn from the environment to achieve their goals. A thermostat is an example of an intelligent agent.

Following are the main four rules for an AI agent:

- **Rule 1:** An AI agent must have the ability to perceive the environment.
- **Rule 2:** The observation must be used to make decisions.
- **Rule 3:** Decision should result in an action.
- **Rule 4:** The action taken by an AI agent must be a rational action.

c. **Uses of Intelligent Agents:**

Intelligent agents have a wide range of applications, including:

- **Problem-solving:** AI agents can be used to solve complex problems in various domains like game playing, scheduling, and scientific discovery.
- **Planning and decision-making:** AI agents can analyze situations, plan courses of action, and make decisions in real-time for autonomous systems like robots or self-driving cars.
- **Learning and adaptation:** AI agents can learn from experience and adapt their behavior to new situations. This is useful for tasks like spam filtering or recommendation systems.

- **Perception and understanding:** AI agents can process sensory data like images or speech to perceive and understand the world around them. This has applications in computer vision, natural language processing, and robotics.
- **Human-computer interaction:** Intelligent agents can be used to create interactive systems that can understand and respond to user queries, like virtual assistants or chatbots.

2. List advantages of Artificial Intelligence.

more powerful and more useful computers

new and improved interface

solving new problems

better handling of information

relieves information overload

conversion of information into knowledge

Reduction in Human Error

Reduce the Risk (Zero Risk)

24/7 Support

Perform Repetitive Jobs

Faster decision

New Inventions

Daily Applications

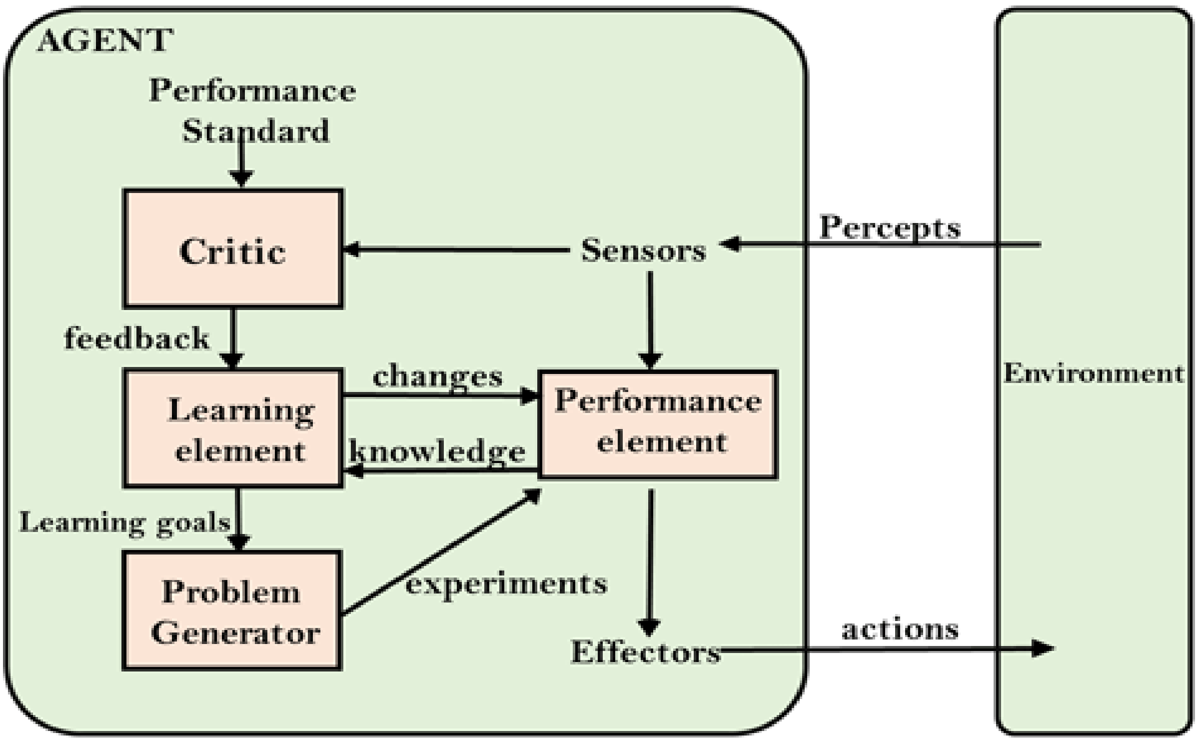
Digital Assistance

AI in risky situations

3. Describe Learning Agent Architecture with diagram.

- A learning agent in AI is the type of agent which can learn from its past experiences, or it has learning capabilities.

- It starts to act with basic knowledge and then able to act and adapt automatically through learning.
- A learning agent has mainly four conceptual components, which are:
    a. **Learning element:** It is responsible for making improvements by learning from environment
    b. **Critic:** Learning element takes feedback from critic which describes that how well the agent is doing with respect to a fixed performance standard.
    c. **Performance element:** It is responsible for selecting external action
    d. **Problem generator:** This component is responsible for suggesting actions that will lead to new and informative experiences.
- Hence, learning agents are able to learn, analyze performance, and look for new ways to improve the performance.



4. properties of agent task environment

1. Fully observable vs Partially Observable
2. Static vs Dynamic
3. Discrete vs Continuous
4. Deterministic vs Stochastic
5. Single-agent vs Multi-agent
6. Episodic vs sequential
7. Known vs Unknown
8. Accessible vs Inaccessible

1. Fully observable vs Partially Observable:

- If an agent sensor can sense or access the complete state of an environment at each point of time then it is **a fully observable** environment, else it is **partially observable**.

  An agent with no sensors in all environments then such an environment is called as **unobservable**.

2. Deterministic vs Stochastic:

- If an agent's current state and selected action can completely determine the next state of the environment, then such environment is called a deterministic environment.
- A stochastic environment is random in nature and cannot be determined completely by an agent.

3. Episodic vs Sequential:

- In an episodic environment, there is a series of one-shot actions, and only the current percept is required for the action.
- However, in Sequential environment, an agent requires memory of past actions to determine the next best actions.

4. Single-agent vs Multi-agent

- If only one agent is involved in an environment, and operating by itself then such an environment is called single agent environment.
- However, if multiple agents are operating in an environment, then such an environment is called a multi-agent environment.

5. Static vs Dynamic:

- If the environment can change itself while an agent is deliberating then such environment is called a dynamic environment else it is called a static environment.

6. Discrete vs Continuous:

- If in an environment there are a finite number of percepts and actions that can be performed within it, then such an environment is called a discrete environment else it is called continuous environment.

7. Known vs Unknown

- Known and unknown are not actually a feature of an environment, but it is an agent's state of knowledge to perform an action.
- In a known environment, the results for all actions are known to the agent. While in unknown environment, agent needs to learn how it works in order to perform an action.

8. Accessible vs Inaccessible

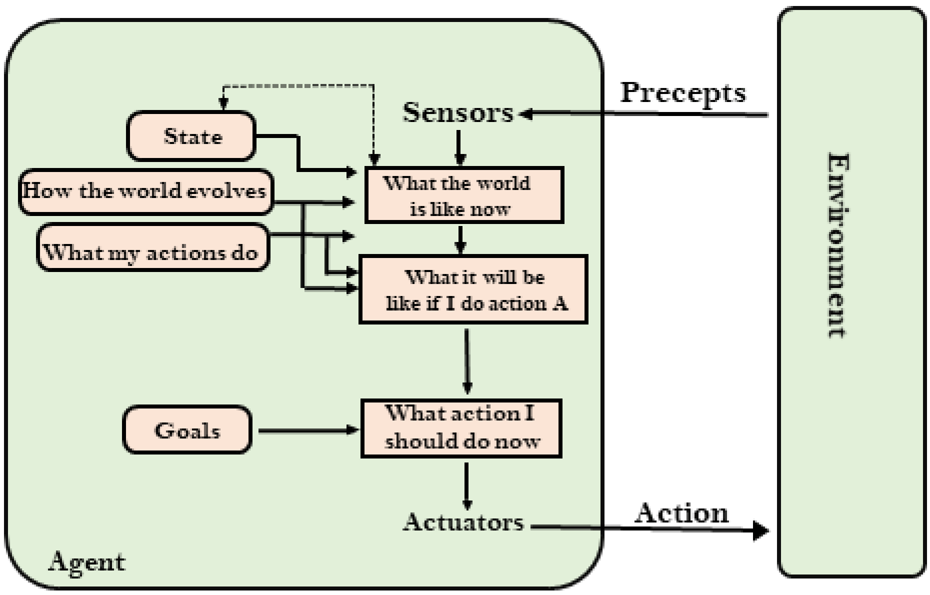
- If an agent can obtain complete and accurate information about the state's environment, then such an environment is called an Accessible environment else it is called inaccessible.

5. goal based agent

Goal-based agents

- The knowledge of the current state environment is not always sufficient to decide for an agent to what to do.
- The agent needs to know its goal which describes desirable situations.

- Goal-based agents expand the capabilities of the model-based agent by having the "goal" information.
- They choose an action, so that they can achieve the goal.
- These agents may have to consider a long sequence of possible actions before deciding whether the goal is achieved or not. Such considerations of different scenario are called searching and planning, which makes an agent proactive.



6. What is rationality and rational agents? Give an example of rational action performed by any intelligent agent.

Rationality :

The rationality of an agent is measured by its performance measure. Rationality can be judged on the basis of following points:

- Performance measure which defines the success criterion.
- Agent prior knowledge of its environment.
- Best possible actions that an agent can perform.
- The sequence of percepts.

Rational Agent :

1. A rational agent is an agent which has clear preferences and models uncertainty via expected values.

2. A rational agent can be anything that makes decisions, typically a person, firm, machine, or software.
3. A rational agent always performs right action, where the right action means the action that causes the agent to be most successful in the given percept sequence.
4. Rational agent is capable of taking best possible action in any situation.

Example of rational action performed by any intelligent agent :

Automated Taxi Driver:

Performance Measure: Safe, fast, legal, comfortable trip, maximize profits.

Environment: Roads, other traffic, customers.

Actuators: Steering wheel, accelerator, brake, signal, horn.

Sensors: Cameras, sonar, speedometer, GPS, odometer, engine sensors, keyboard.

7. Explain the PEAS. Represent PEAS for automated taxi driver and scientific calculator.

Rational Agent: The rational agent considers all possibilities and chooses to perform a highly efficient action. For example, it chooses the shortest path with low cost for high efficiency. **PEAS** stands for a *Performance measure*, *Environment*, *Actuator*, *Sensor*.

- **Performance:** The measure of success for the agent. It's evaluated based on how well the agent achieves its goals within its environment.
- **Environment:** The world or surroundings in which the agent operates. It includes everything the agent can perceive and interact with.
  There are 5 major types of environments:
  - Fully Observable & Partially Observable
  - Episodic & Sequential
  - Static & Dynamic
  - Discrete & Continuous
  - Deterministic & Stochastic
- **Actuators:** The system itself, including its internal state, decision-making functions, and capabilities.
- **Sensors:** The mechanisms through which the agent gathers information about the environment.

a. Automated Taxi Driver :

Performance Measure: Safe, fast, legal, comfortable trip, Optimum speed

Environment: Roads, other traffic, City streets, highways, vehicle

Actuators: Steering wheel, accelerator, brake, signal, horn.

Sensors: Cameras, sonar, speedometer, GPS, odometer, engine sensors, keyboard.

b. scientific calculator :

Performance: Accurately perform mathematical calculations according to user input and mathematical functions.

Environment: User interaction, mathematical symbols and expressions.

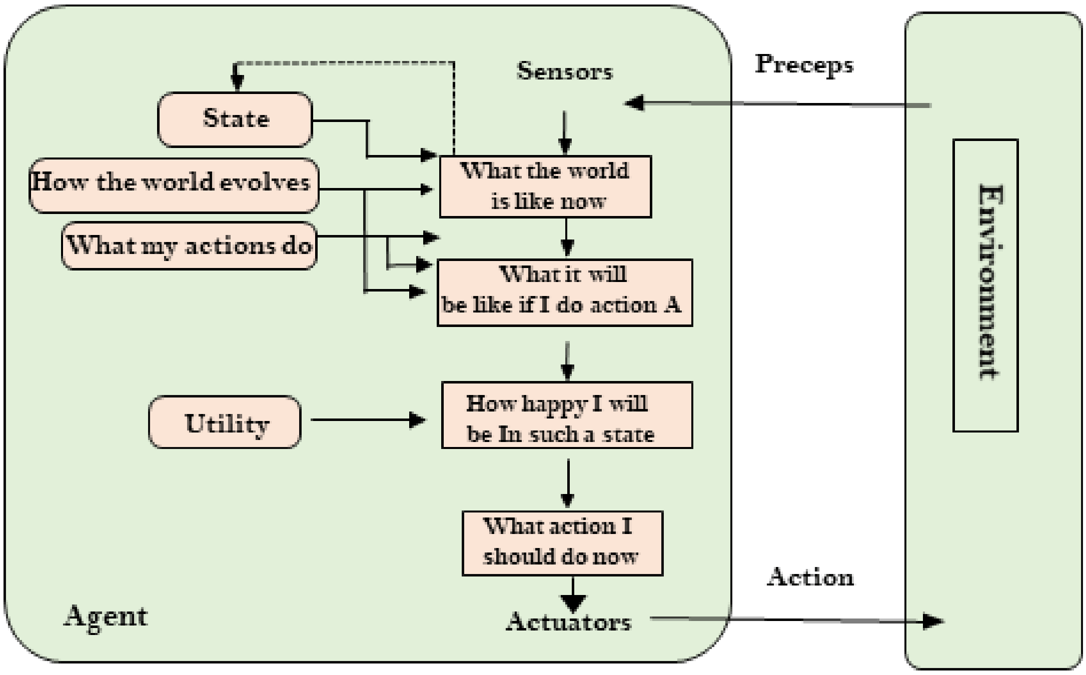
Actuators: The scientific calculator program.

Sensors: Keyboard or touchscreen for user input.

8.Elaborate Utility based agent and Learning agent with the help of neat diagram.

Utility-based agents

- These agents are similar to the goal-based agent but provide an extra component of utility measurement which makes them different by providing a measure of success at a given state.
- Utility-based agent act based not only goals but also the best way to achieve the goal.
- The Utility-based agent is useful when there are multiple possible alternatives, and an agent has to choose in order to perform the best action.
- The utility function maps each state to a real number to check how efficiently each action achieves the goals.



9.Define Artificial Intelligence and elaborate the applications of artificial intelligence in real world.

. **Artificial Intelligence** is composed of two words **Artificial** and **Intelligence**, where Artificial defines "*man-made,*" and intelligence defines "*thinking power*", hence AI means "*a man-made thinking power.*"

"It is a branch of computer science by which we can create intelligent machines which can behave like a human, think like humans, and able to make decisions."

Artificial Intelligence exists when a machine can have human based skills such as learning, reasoning, and solving problems.

With Artificial Intelligence you do not need to preprogram a machine to do some work, despite that you can create a machine with programmed algorithms which can work with own intelligence, and that is the awesomeness of AI.

AI in Astronomy

- Artificial Intelligence can be very useful to solve complex universe problems. AI technology can be helpful for understanding the universe such as how it works, origin, etc.

2. AI in Healthcare

- In the last, five to ten years, AI becoming more advantageous for the healthcare industry and going to have a significant impact on this industry.
- Healthcare Industries are applying AI to make a better and faster diagnosis than humans. AI can help doctors with diagnoses and can inform when patients are worsening so that medical help can reach to the patient before hospitalization.

3. AI in Gaming

- AI can be used for gaming purpose. The AI machines can play strategic games like chess, where the machine needs to think of a large number of possible places.

4. AI in Finance

- AI and finance industries are the best matches for each other. The finance industry is implementing automation, chatbot, adaptive intelligence, algorithm trading, and machine learning into financial processes.

5. AI in Data Security

- The security of data is crucial for every company and cyber-attacks are growing very rapidly in the digital world. AI can be used to make your data more safe and secure. Some examples such as AEG bot, AI2 Platform,are used to determine software bug and cyber-attacks in a better way.

6. AI in Social Media

- Social Media sites such as Facebook, Twitter, and Snapchat contain billions of user profiles, which need to be stored and managed in a very efficient way. AI can organize and manage massive amounts of data. AI can analyze lots of data to identify the latest trends, hashtag, and requirement of different users.

7. AI in Travel & Transport

- AI is becoming highly demanding for travel industries. AI is capable of doing various travel related works such as from making travel arrangement to suggesting the hotels, flights, and best routes to the customers. Travel industries are using AI-powered chatbots which can make human-like interaction with customers for better and fast response.

8. AI in Automotive Industry

- Some Automotive industries are using AI to provide virtual assistant to their user for better performance. Such as Tesla has introduced TeslaBot, an intelligent virtual assistant.
- Various Industries are currently working for developing self-driven cars which can make your journey more safe and secure.

9. AI in Robotics:

- Artificial Intelligence has a remarkable role in Robotics. Usually, general robots are programmed such that they can perform some repetitive task, but with the help of AI, we can create intelligent robots which can perform tasks with their own experiences without pre-programmed.
- Humanoid Robots are best examples for AI in robotics, recently the intelligent Humanoid robot named as Erica and Sophia has been developed which can talk and behave like humans.

10. AI in Entertainment

- We are currently using some AI based applications in our daily life with some entertainment services such as Netflix or Amazon. With the help of ML/AI algorithms, these services show the recommendations for programs or shows.

11. AI in Agriculture

- Agriculture is an area which requires various resources, labor, money, and time for best result. Now a day's agriculture is becoming digital, and AI is emerging in this field. Agriculture is applying AI as agriculture robotics, solid and crop monitoring, predictive analysis. AI in agriculture can be very helpful for farmers.

12. AI in E-commerce

- AI is providing a competitive edge to the e-commerce industry, and it is becoming more demanding in the e-commerce business. AI is helping shoppers to discover associated products with recommended size, color, or even brand.

13. AI in education:

- AI can automate grading so that the tutor can have more time to teach. AI chatbot can communicate with students as a teaching assistant.
- AI in the future can be work as a personal virtual tutor for students, which will be accessible easily at any time and any place.

10.Explain Deterministic environment and episodic environment with example.

Deterministic Environment:

- If an agent's current state and selected action can completely determine the next state of the environment, then such environment is called a deterministic environment.
- A deterministic environment is one in which the outcome of an action is completely predictable and can be precisely determined.
- The state of the environment completely determines the result of an agent's action, an a deterministic environment.

Example:

- Consider the game of Tic-Tac-Toe.
- The environment (game board) is deterministic.
- If the agent (player X) marks the top left corner (current state), and the action is to place an "X" in the center (action), the next state will always be the game board with an "X" in the center.

episodic environment :

- In an episodic environment, there is a series of one-shot actions, and only the current percept is required for the action.
- Each episode starts with an initial state, followed by a sequence of agent actions and the corresponding environmental responses (percepts) until a terminal state is reached.
- Once the terminal state is reached, the episode ends, and a new episode might begin with a new initial state.
- In episodic environments, the agent's decisions in a particular episode typically have no impact on future episodes (except for learning agents that might accumulate knowledge across episodes)

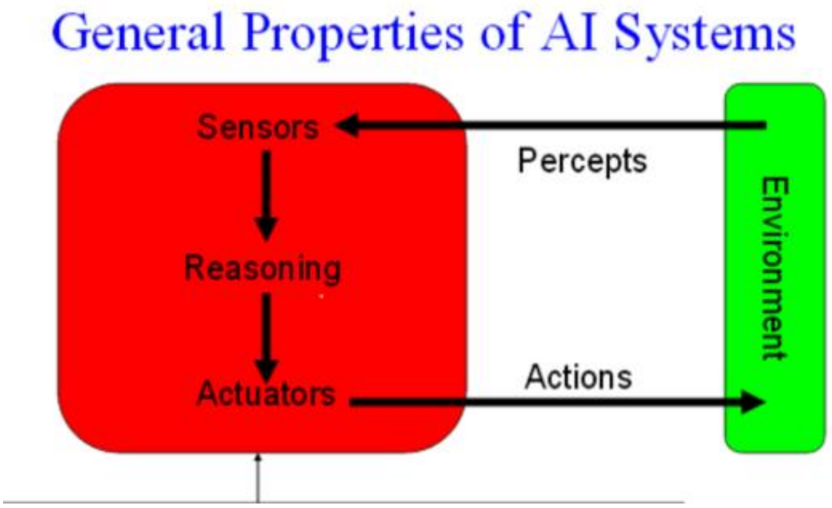
Example:

- A robot vacuuming a room can be considered an episodic environment.
- Each cleaning session is an episode.
- The initial state is the dirty room.
- The agent (robot) takes actions like moving forward, turning, and vacuuming.
- The environment provides percepts about bumps, dirt detection, etc.

- The terminal state is a clean room.
- Once clean, a new episode might begin when the room gets dirty again.

11. Explain an Agent and agent program with example.

Agents :-



An agent perceives its environment through sensors -

1.the complete set of inputs at a given time is called a percept

2.the current percept, or a sequence of percepts may influence the actions of an agent

It can change the environment through actuators -

1.an operation involving an actuator is called an action

2.actions can be grouped into action sequences

Ex. Human agent -

eyes, ears, skin, taste buds, etc. for sensors

hands, fingers, legs, mouth, etc. for actuators

Agent program:

Implements the agent function, running on agents architecture.

Here's a breakdown of the core components:

1. **Sensors:** Gather information about the environment (percepts).
2. **Effectors:** Perform actions that modify the environment.
3. **Actuators:** Physical mechanisms that carry out the actions chosen by the program. (May not be present in all agent programs)
4. **Internal State:** Represents the agent's knowledge, beliefs, and goals.

5. **Decision-Making Function:** Uses the internal state and percepts to choose actions.

Example:

Consider a simple **robot vacuum cleaner** as an agent:

- **Environment:** A house with dust and dirt.
- **Sensors:** Bumpers, dust sensors, cliff sensors.
- **Effectors:** Wheels, vacuum cleaner motor.
- **Internal State:** Battery level, location map (optional).
- **Decision-Making Function:** A set of rules (e.g., move forward until bump sensor activated, then turn).

12. Foundations and History of Artificial Intelligence

Foundation of ai -

1. machine learning**: Machine learning is a subset of AI that focuses on the development of algorithms that enable** machines to learn from data and make predictions **or decisions without being explicitly programmed.**

2. natural language Processing: Natural Language Processing (NLP) refers to the ability of machines to understand, interpret, and generate human language, chatbot

3. computer vision**: Computer Vision involves training machines to interpret and understand visual data from the world around them. With computer vision, machines can recognize objects, faces, and even emotions, which is critical for applications such as facial recognition, surveillance, and self-driving cars.**

4. Robotics: Robotics is the application of AI in the development of robots that can perform tasks autonomously. This includes everything from industrial robots used in manufacturing to autonomous drones and self-driving cars.

5.**Philosophy =** Rationalism, Dualism, Materialism, Empiricism, Induction, Logical Positivism, Confirmation Theory.

6.Mathematics = Algorithm, incompleteness theorem, computable, tractability, NP completeness, Non deterministic polynomial and probability.

7.Economics = Utility, Decision Theory, Game Theory, Operations Research.

8. Neuroscience = Neuroscience is the study of the nervous system, especially the brain.

9. Psychology = Behaviourism, Cognitive psychology.

10. Linguistics = Computational linguistics or natural language processing and knowledge representation.

History of ai –

Maturation of Artificial Intelligence (1943-1952)

The birth of Artificial Intelligence (1952-1956)

The golden years-Early enthusiasm (1956-1974)

The first AI winter (1974-1980)

A boom of AI (1980-1987)

The second AI winter (1987-1993)

The emergence of intelligent agents (1993-2011)

Deep learning, big data and artificial general intelligence (2011 - present)

Unit 2

1.Describe Hill climbing search with suitable example. List disadvantage of hill climbing process.

- Hill climbing algorithm is a local search algorithm which continuously moves in the direction of increasing elevation/value to find the peak of the mountain or best solution to the problem. It terminates when it reaches a peak value where no neighbor has a higher value.
- Hill climbing algorithm is a technique which is used for optimizing the mathematical problems. One of the widely discussed examples of Hill climbing algorithm is Traveling-salesman Problem in which we need to minimize the distance traveled by the salesman.
- It is also called greedy local search as it only looks to its good immediate neighbor state and not beyond that.
- A node of hill climbing algorithm has two components which are state and value.
- Hill Climbing is mostly used when a good heuristic is available.
- In this algorithm, we don't need to maintain and handle the search tree or graph as it only keeps a single current state.

Types of Hill Climbing Algorithm :

- Simple hill Climbing
- Steepest-Ascent hill-climbing
- Stochastic hill Climbing

Limitations (Drawbacks in AI):

- Local maxima: This is a state better than the local region or neighboring states, but not a global maxima. This occurs since a better solution exit which is not in the vicinity of the present state.
- Plateau: This refers to a flat area or space where neighborhood states have the same value as the present state and hence fail to determine the best direction to move on.
- Ridge: It is a search space at a higher attitude than the surroundings. It cannot be traversed by single move. It only moves in one direction doesn’t change direction.

Traveling Salesperson

States → location / cities
↳ States
- each city may be visited only once.
- visited cities must be kept as state informat^{n}

initial State → Starting point
no cities visited

successor function (operators) →
move from 1 locat^{n} to another 1.

Goal test → all locat^{n} visited
agent at the initial loca^{n}

Path Cost → distance bet^{n} locations.

2. Differentiate between uninformed and informed search. Show an example for each. Search technique.

Parameters	Informed Search	Uninformed Search
Utilizing Knowledge	It uses knowledge during the process of searching.	It does not require using any knowledge during the process of searching.
Speed	Finding the solution is quicker.	Finding the solution is much slower comparatively.
Completion	It can be both complete and incomplete.	It is always bound to be complete.
Consumption of Time	Due to a quicker search, it consumes much less time.	Due to slow searches, it consumes comparatively more time.
Cost Incurred	The expenses are much lower.	The expenses are comparatively higher.
Suggestion/ Direction	The AI gets suggestions regarding how and where to find a solution to any problem.	The AI does not get any suggestions regarding what solution to find and where to find it. Whatever knowledge it gets is out of the information provided.
Efficiency	It costs less and generates quicker results. Thus, it is comparatively more efficient.	It costs more and generates slower results. Thus, it is comparatively less efficient.
Length of Implementation	Implementation is shorter using AI.	The implementation is lengthier using AI.
Examples	A few examples include Graph Search and Greedy Search.	A few examples include Breadth-First Search or BFS and Depth-First Search or DFS.

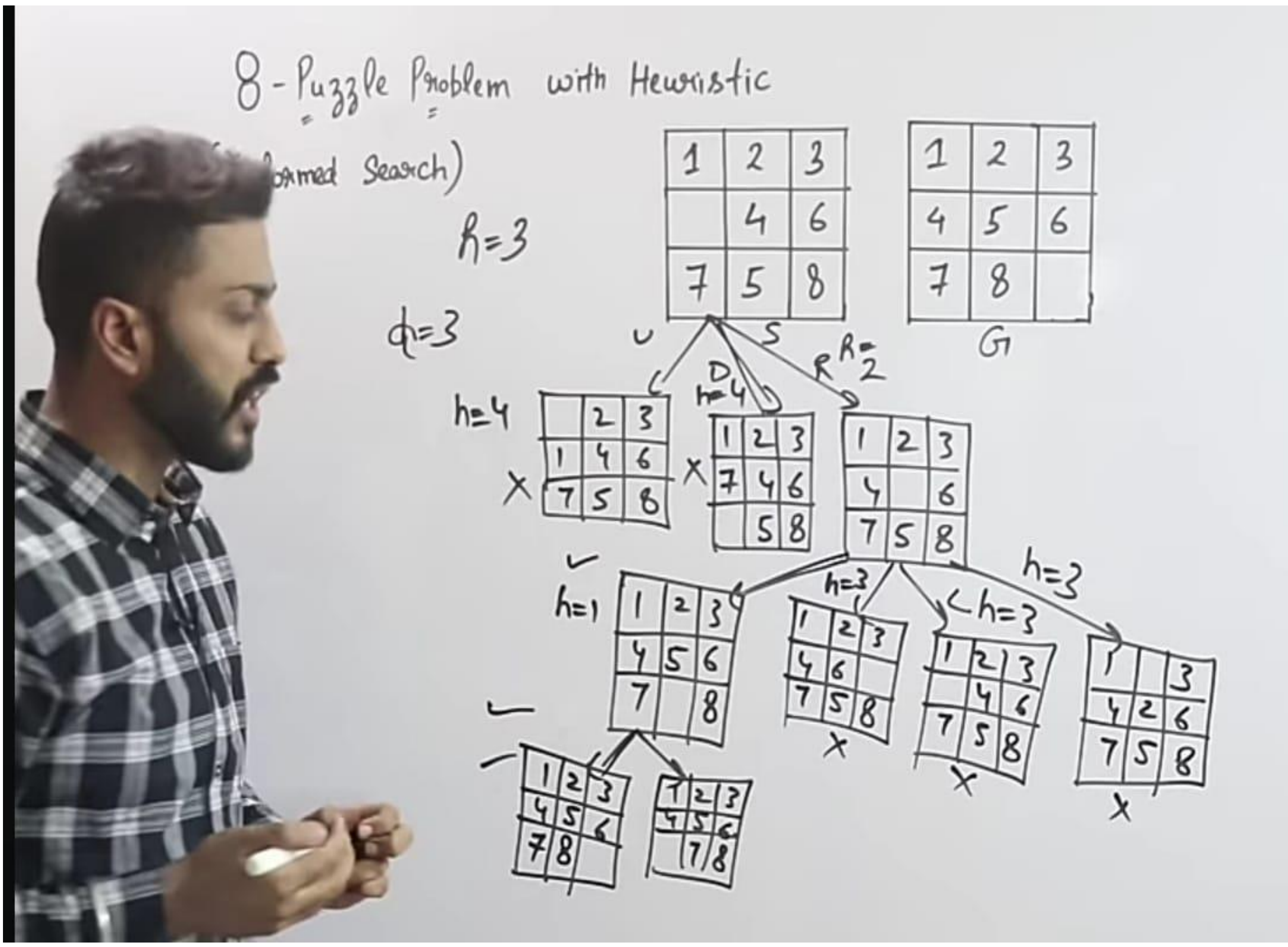
3. Explain the concept of a heuristic function with an example.

- A heuristic function (h(n)) is a function that estimates the cost (distance) of reaching the goal state from a particular state (n) in a problem.
- It provides an approximate measure of how "close" a state is to the goal.

- A heuristic is a technique that is used to solve a problem faster than the classic methods. These techniques are used to find the approximate solution of a problem when classical methods do not. Heuristics are said to be the problem-solving techniques that result in practical and quick solutions.
- A heuristic function in artificial intelligence, also known as a heuristic or simply a heuristic, is an evaluation function used to estimate the cost or potential of reaching a goal state from a given state in a problem-solving domain.
- Heuristics are used to make informed decisions in situations where it's computationally expensive to search through all possible states or actions.
- Best first search, greedy search, A* algorithm and hill climbing search this all informed search algorithms u
- ses heuristics.

Example :

Heuristic Evaluation for 8-puzzle problem in AI



4.Explain different search strategies

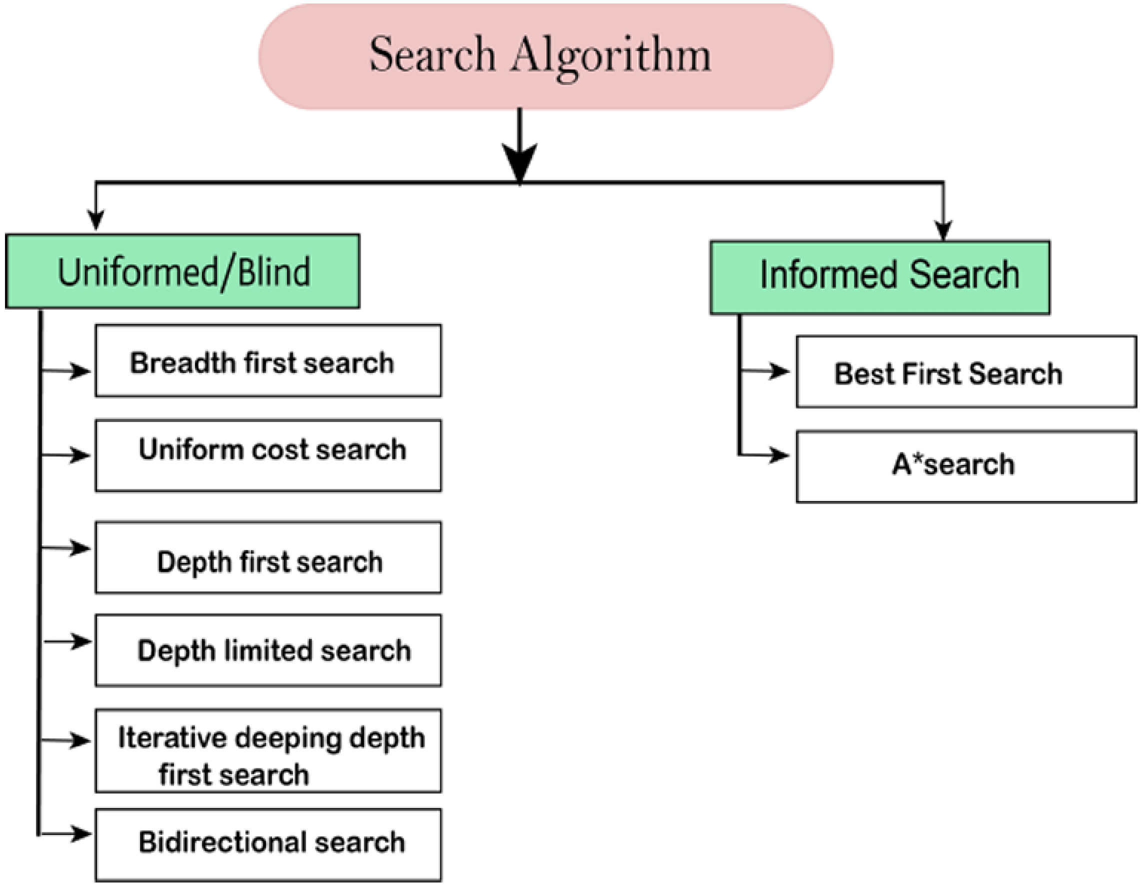


Fig: Search algorithm

Uninformed search strategies –

1. Breadth first search
2. Depth first search

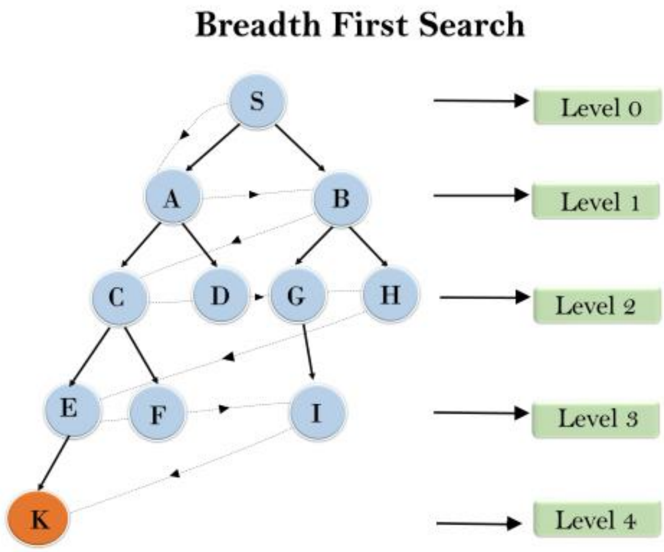
Informed search strategies –

1. A* search

1.Breadth first search –

- Breadth-first search is the most common search strategy for traversing a tree or graph. This algorithm searches breadthwise in a tree or graph, so it is called breadth-first search.
- BFS algorithm starts searching from the root node of the tree and expands all successor node at the current level before moving to nodes of next level.
- The breadth-first search algorithm is an example of a general-graph search algorithm.

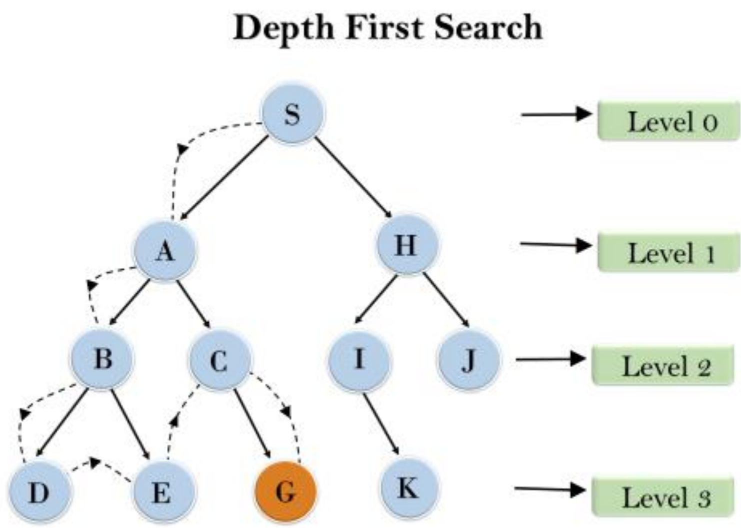
- Breadth-first search implemented using FIFO queue data structure.



S---> A--->B---->C--->D---->G--->H--->E---->F---->I---->K

2.Depth first search –

- Depth-first search is a recursive algorithm for traversing a tree or graph data structure.
- It is called the depth-first search because it starts from the root node and follows each path to its greatest depth node before moving to the next path.
- DFS uses a stack data structure for its implementation.
- The process of the DFS algorithm is similar to the BFS algorithm.



S---> H--->J---->I--->K---->A--->C--->G---->B---->E---->D

3.A* search –

A* Search algorithm is one of the best and popular technique used in path-finding and graph traversals.

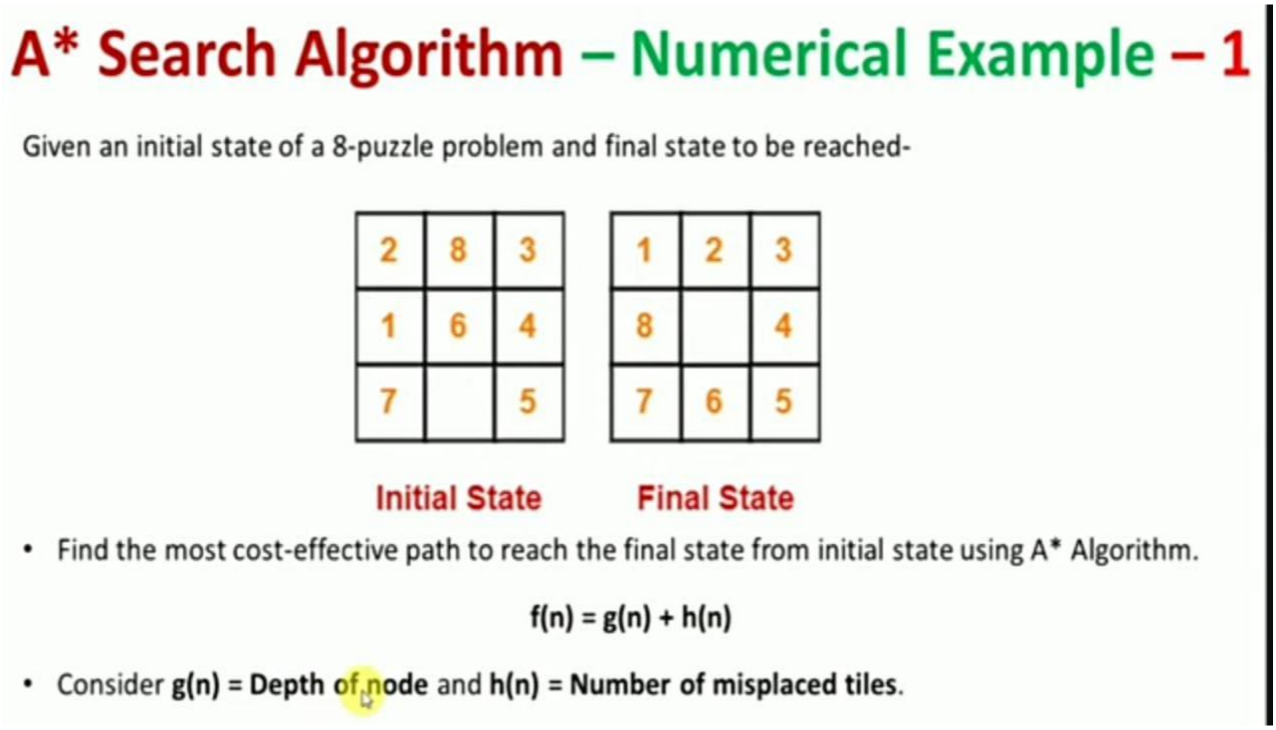
A* algorithm is that it keeps a track of each visited node which helps in ignoring the nodes that are already visited, saving a huge amount of time.

By using A* search algorithm , we find cost effective path from the initial state to final state

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

A* uses a combination of heuristic value (h-score: how far the goal node is) as well as the g-score (i.e. the number of nodes traversed from the start node to current node).

The next node chosen from the open list is based on its **f score**, the node with the least f score is picked up and explored.



5.Explain Depth first search and the concept of Depth First Iterative Deepening search with an example.

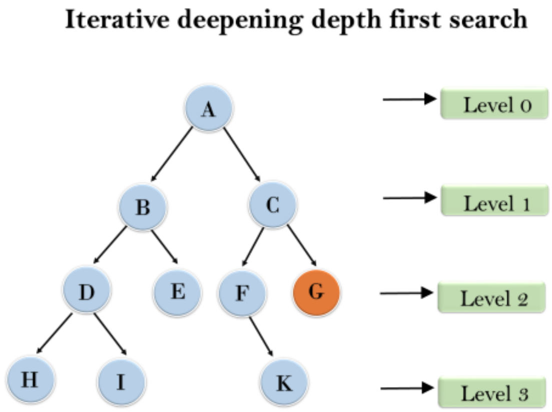
Depth First Iterative Deepening search

The iterative deepening algorithm is a combination of DFS and BFS algorithms. This search algorithm finds out the best depth limit and does it by gradually increasing the limit until a goal is found.

This algorithm performs depth-first search up to a certain "depth limit", and it keeps increasing the depth limit after each iteration until the goal node is found.

This Search algorithm combines the benefits of Breadth-first search's fast search and depth-first search's memory efficiency.

The iterative search algorithm is useful uninformed search when search space is large, and depth of goal node is unknown.



1'st Iteration-----> A
2'nd Iteration----> A, B, C
3'rd Iteration------>A, B, D, E, C, F, G
4'th Iteration------>A, B, D, H, I, E, C, F, K, G
In the fourth iteration, the algorithm will find the goal node.

6. Evaluate the steps in a simple problem-solving technique with a relevant example.

Steps for Problem Solving in AI

1. **Define the problem:**
    - Initial state: Identify the starting condition or configuration.
    - Goal state: Clearly state what the problem solver is trying to achieve.
    - Relevant constraints or operators: List any limitations or rules that must be followed, as well as actions that can be taken to change states.
2. **Formulate a search space:**
    - Determine all possible states that can be reached from the initial state using the defined operators.
    - Create a representation of the search space, such as a graph or tree, to visualize the possible states and transitions.
3. **Define a search strategy:**

- Choose an appropriate algorithm to explore the search space effectively.
- Blind search strategies:
    - Breadth-First Search (BFS): Explores all neighbor nodes at the present depth prior to moving on to the nodes at the next depth level.
    - Depth-First Search (DFS): Explores as far as possible along each branch before backtracking.
- Informed search strategies:
    - A* Search: Utilizes a heuristic function to estimate the cost of reaching the goal from any given state, guiding the search towards the most promising states.

4. **Apply the search strategy:**
    - Implement the chosen algorithm using the defined search space and operators.
    - Execute the algorithm to explore the search space and identify a sequence of actions or operators that lead from the initial state to the goal state.
5. **Solution validation:**
    - Verify if the identified sequence of actions or operators satisfies the problem's constraints and reaches the desired goal state.
    - Test the solution using sample inputs or scenarios to ensure correctness and efficiency.
    - Refine the solution if necessary based on validation results.

Let's apply these problem-solving steps to a classic problem: the "Tower of Hanoi" puzzle.

1. **Define the problem:**
    - Initial state: Three pegs and a set of disks of different sizes stacked in ascending order of size on one peg, with the largest disk at the bottom and the smallest on top.
    - Goal state: Move all disks from the initial peg to another peg, while obeying the rule that no larger disk can be placed on top of a smaller disk.
    - Relevant constraints or operators: The only allowed operation is to move one disk at a time from the top of one peg to the top of another peg, with the constraint that no disk may be placed on top of a smaller disk.
2. **Formulate a search space:**
    - The search space consists of all possible configurations of disks on the three pegs.

- Each state represents a different arrangement of disks on the pegs.

3. **Define a search strategy:**
   - A suitable algorithm for this problem is Depth-First Search (DFS), as it explores each branch of the search space fully before moving on to the next branch.
   - The search space is relatively small and finite for the Tower of Hanoi problem, so DFS can efficiently find the solution.
4. **Apply the search strategy:**
   - Implement DFS to explore the search space by recursively moving disks from one peg to another, while obeying the rules of the puzzle.
   - At each step, evaluate possible moves and continue until reaching the goal state where all disks are stacked on a different peg.
5. **Solution validation:**
   - Verify if the final configuration of disks on the pegs adheres to the rules of the puzzle.
   - Check that all disks are moved from the initial peg to the target peg.
   - Test the solution with different numbers of disks to ensure it works efficiently for various problem sizes.

8.Describe problem representation and analysis with example of robot navigation.

- **Initial State**: The initial state that the agent starts in.
- **Action**: Description of possible actions available to the agent.
- **Successor function** : It is a description of possible actions, a set of operators. This function returns a set of <action, successor>ordered pair.
- **State Space**: The set of all states reachable from the initial state. State space forms a graph in which nodes are states and arcs between nodes are actions.
- **Path** : A path in the state space is a sequence of states connected by a sequence of actions.
- **Goal Test**: It determines whether a given state is goal state.
- **Path cost**: It is a function that assigns a numeric cost to each path.

  (Cost of path can be described as the sum of the cost of the individual actions along the path.)

- **Solution**: A path from the initial state to a goal state.

Example - Robot Navigation

States -

locations

position of actuators

initial state -

start position (dependent on the task)

successor function (operators) -

movement, actions of actuators

goal test -

task-dependent

path cost -

may be very complex

-distance, energy consumption

9.Elaborate on Problem Reduction Methods.

Problem reduction is a powerful technique by using it we can find solution for complex problems. By dividing that problem into sub problems, and by solving that sub problems individually we get solution of the original problem.

Steps involved :

Decomposing the Problem: This is the heart of problem reduction. You analyze the complex problem and identify its key components or sub-problems. These sub-problems should be independent (meaning solving one doesn't necessarily affect the others) and collectively address the entire original problem.

Solving Sub-problems: Once you have your sub-problems, you can apply various AI techniques or algorithms to solve them individually. This might involve using existing knowledge bases, search algorithms, or specialized algorithms designed for specific sub-problem types.

Combining Solutions: After solving the sub-problems, you need to combine their solutions to get the solution for the original problem. This might involve merging results, making choices based on the sub-problem solutions, or following a specific order of applying sub-problem solutions.

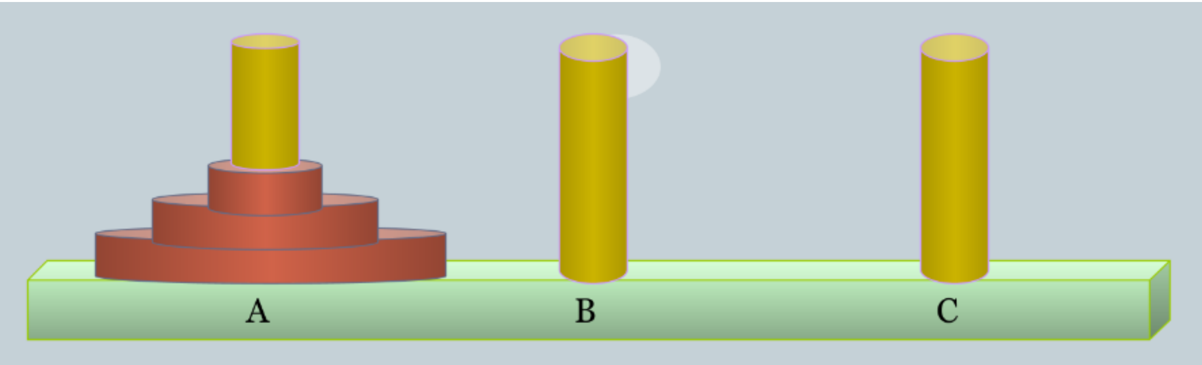
Common Problem Reduction Techniques:

Divide and Conquer: This is a classic example where the problem is broken down into smaller, similar sub-problems. Sorting algorithms like Merge Sort often employ this technique.

Means-Ends Analysis: Here, the current state is compared to the desired goal state, and the AI identifies the difference (the "means"). Then, it breaks down bridging that gap into sub-problems. This is useful in planning and robotics.

-In a problem reduction space, the nodes represent problems to be solved or goals to be achieved, and the edges represent the decomposition of the problem into subproblems.

-This is best illustrated by the example of the Towers of Hanoi problem.



-The root node, labeled “3AC” represents the original problem of transferring all 3 disks from peg A to peg C.

-The goal can be decomposed into three subgoals: 2AB, 1AC, 2BC. In order to achieve the goal, all 3 subgoals must be achieved.