CAP 378
ARTIFICIAL INTELIGENCE

Tanzeela Javid Kaloo (32638)
Assistant Professor
System And Architecture
Lovely Professional University

UNIT – I
Introduction of Problem Solving

Tanzeela Javid Kaloo (32638)
Assistant Professor
System And Architecture
Lovely Professional University

Content

- Introduction To Artificial Intelligence
- Types Of Intelligence
- General Vs Narrow AI
- Strong Vs Weak AI
- Difference Between Human And Machine Intelligence
- Applications Of AI
- Approaches Of Artificial Intelligence
- Intelligent Agent
- Types Of Agent Programs
- Formulating Problems -
    - Water Jug Problem
    - 8 Puzzle Problem
    - Missionaries And Cannibals Problem

What is Intelligence? Philosophical Perspectives

- **Defining Intelligence:**
  - Intelligence is the capacity to learn, adapt, reason, and solve problems in a dynamic environment.
  - It also includes creativity, emotional understanding, and social skills.
- **Philosophical Roots:**
  - **Ancient Greece:** Aristotle's concept of logic and reasoning as the foundation of intelligence.
  - **Enlightenment Era:** Descartes' philosophy emphasizing rational thought ("I doubt, therefore I think, therefore I am").
  - **Modern Science:** Intelligence as the measurable capacity for cognition and learning.

How is intelligence defined across disciplines like philosophy, psychology, and AI?

- **Philosophy** emphasizes the abstract, metaphysical aspects of intelligence, such as consciousness and reasoning.
- **Psychology** focuses on measurable cognitive and emotional abilities, aiming to understand human intelligence in practical and scientific terms.

What distinguishes human intelligence from artificial systems?

What distinguishes human intelligence from artificial systems?

AI defines intelligence functionally, aiming to replicate or simulate intelligent behaviors through algorithms and systems.

Artificial Intelligence

- **Turing Test**: To evaluate whether a machine can exhibit behavior indistinguishable from that of a human in a conversational setting.
- **Functional Definition**: The focus is on **outcomes** rather than mimicking the biological processes of the human brain.
- **Types of AI Intelligence**
  - **Reactive Intelligence**: Reactive systems lack memory and do not "learn" over time.
  - **Adaptive Intelligence**: Utilizes memory to update models over time.
  - **General Intelligence**: Can generalize knowledge across domains.

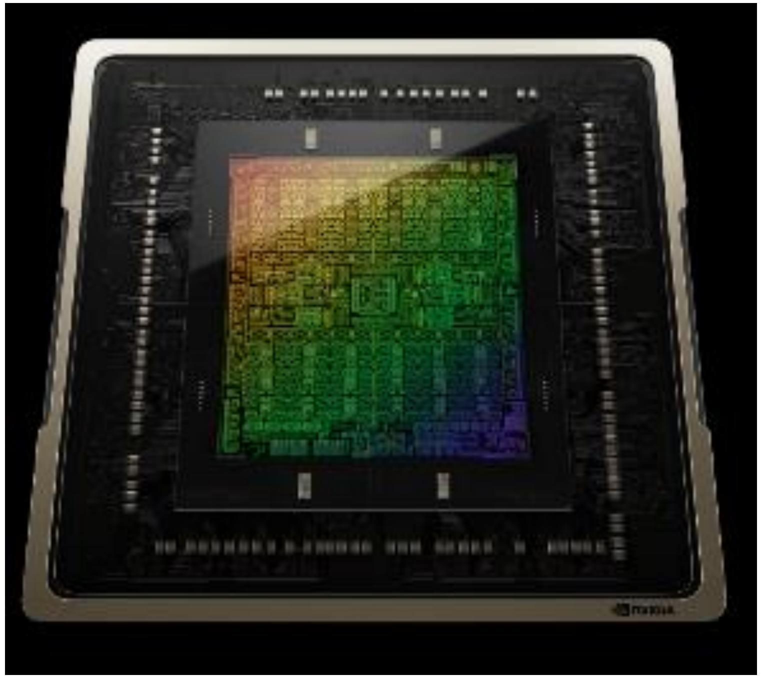
Lady Ada Lovelace
(1815 - 1852) - English Mathematician

First Computer Programmer

An associate of **Charles Babbage**, for whose prototype of a digital computer she created a program.

NVIDIA Ada Lovelace Architecture

Designed to deliver outstanding gaming and creating, professional graphics, AI, and compute performance.



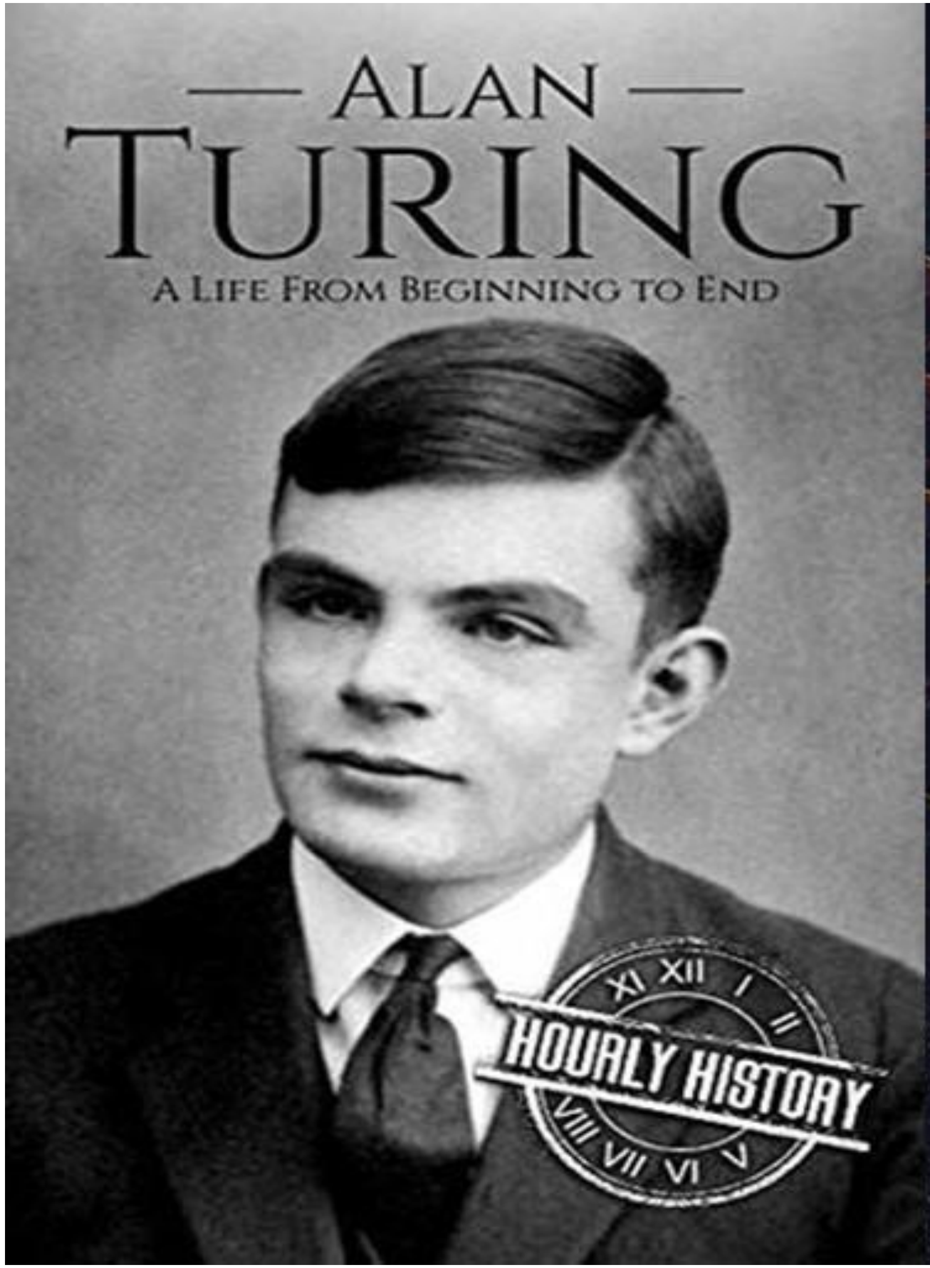
Computing Machinery and Intelligence

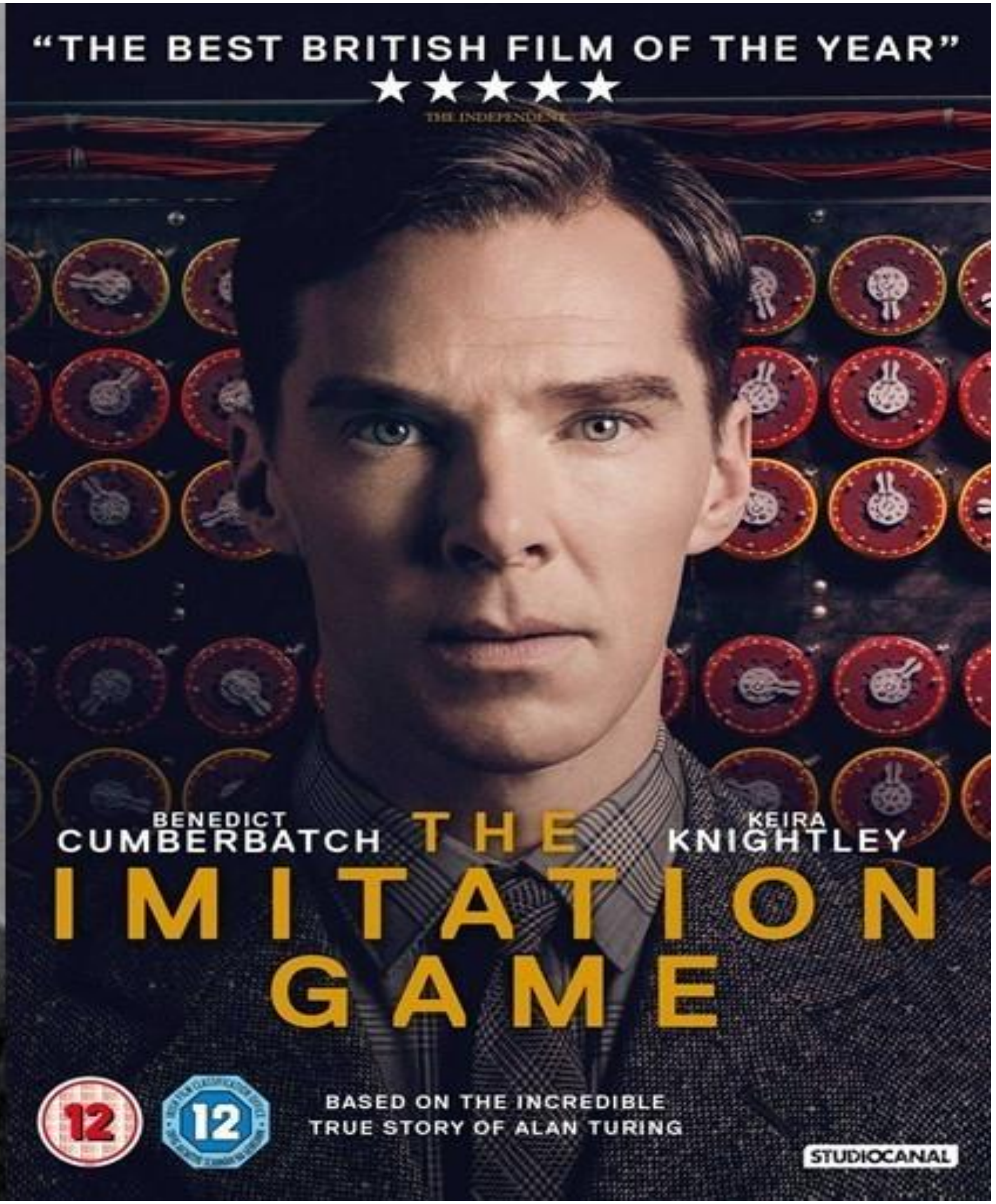
A. M. Turing

1950

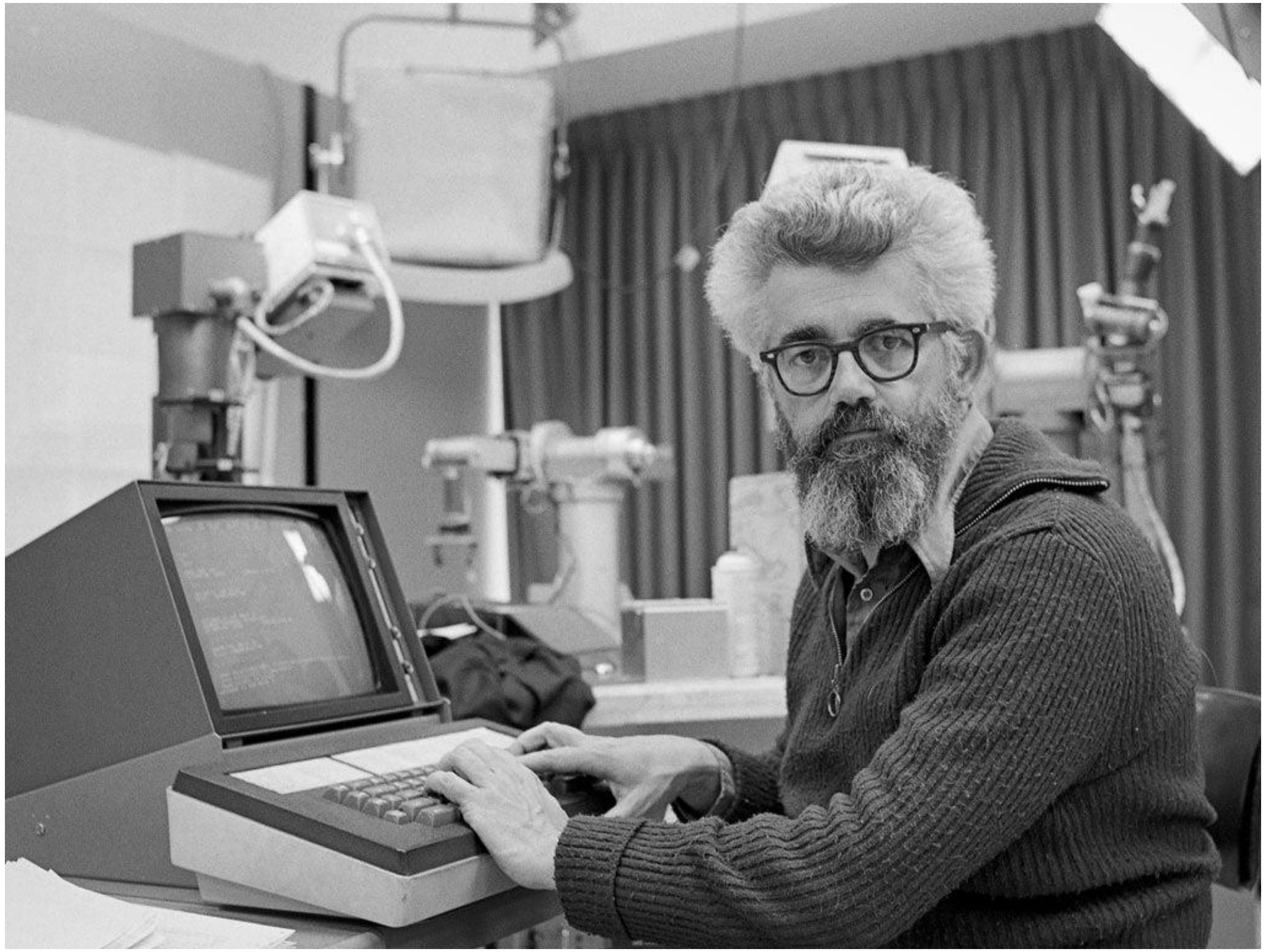
1 The Imitation Game

I propose to consider the question, "Can machines think?" This should begin with definitions of the meaning of the terms "machine" and "think." The definitions might be framed so as to reflect so far as possible the normal use of the words, but this attitude is dangerous, If the meaning of the words "machine" and "think" are to be found by examining how they are commonly used it is difficult to escape the conclusion that the meaning and the answer to the question, "Can machines think?" is to be sought in a statistical survey such as a Gallup poll. But this is absurd. Instead of attempting such a definition I shall replace the question by another, which is closely related to it and is expressed in relatively unambiguous words.

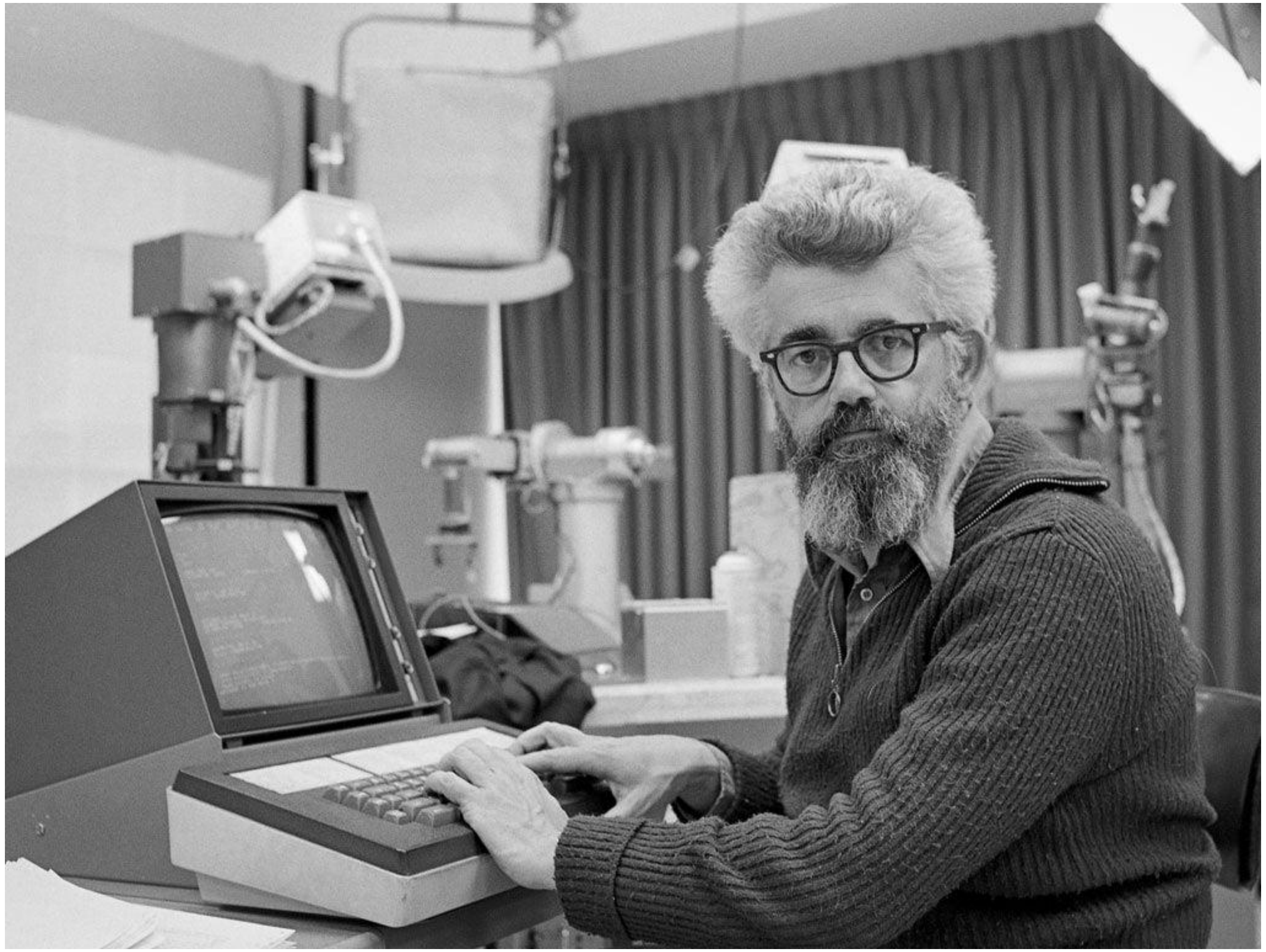

ALAN
TURING
A LIFE FROM BEGINNING TO END
HOURLY HISTORY


"THE BEST BRITISH FILM OF THE YEAR"
★★★★★
BENEDICT
CUMBERBATCH
KEIRA
KNIGHTLEY
THE
IMITATION
GAME
12
12
BASED ON THE INCREDIBLE
TRUE STORY OF ALAN TURING
STUDIOCANAL

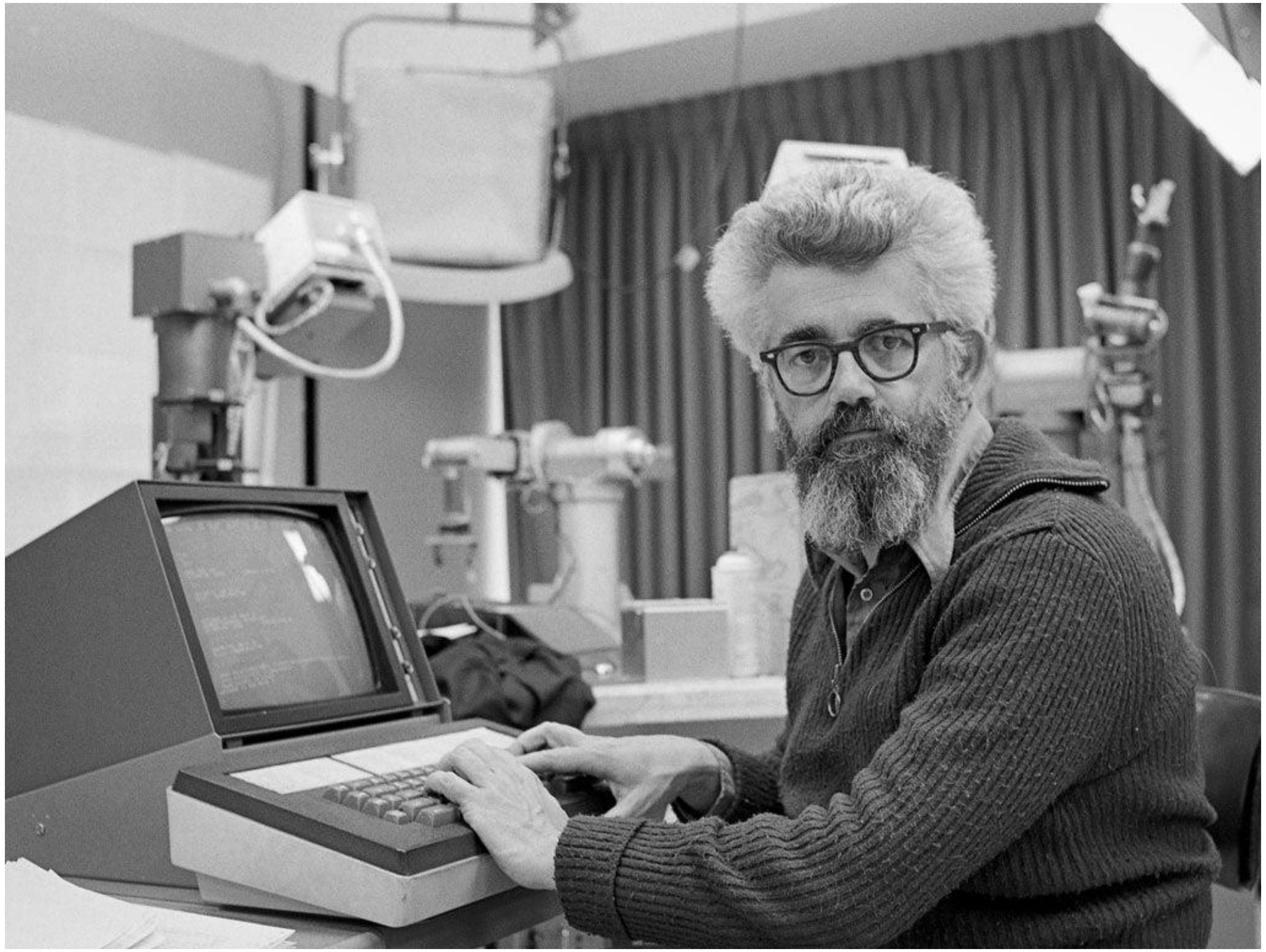
American Computer Scientist



He Coined term for AI (1955)



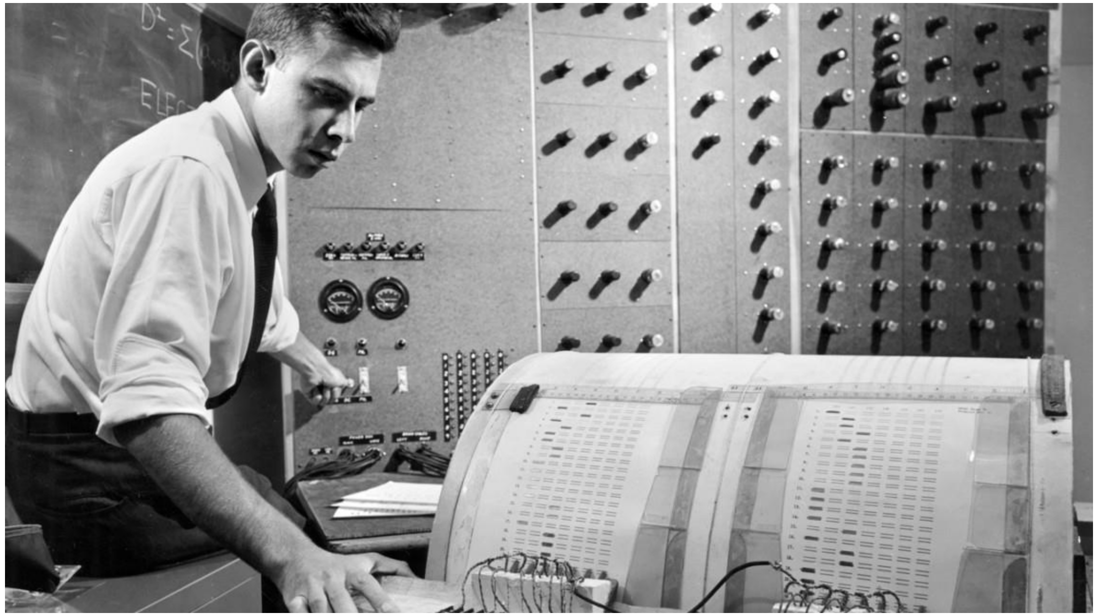
John McCarthy: Computer scientist known as the father of AI



An IBM 704 – a 5-ton computer the size of a room – was fed a series of punch cards. After 50 trials, the computer taught itself to distinguish cards marked on the left from cards marked on the right.

It was a binary classifier that learned to distinguish between two categories, such as shapes.

Professor Frank Rosenblatt



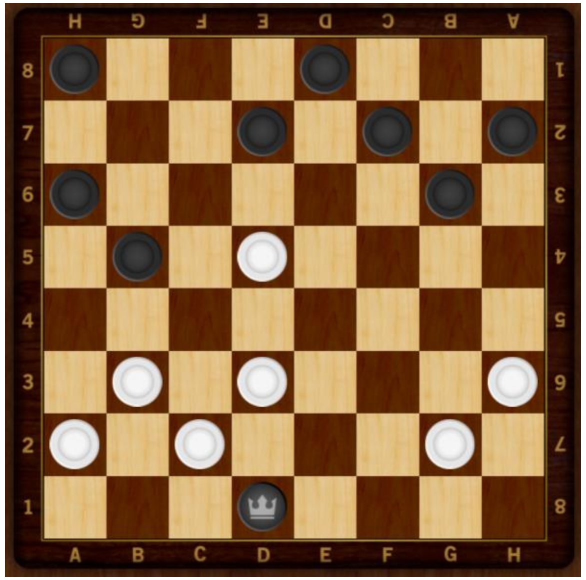
The perceptron, introduced by Frank Rosenblatt in 1958.

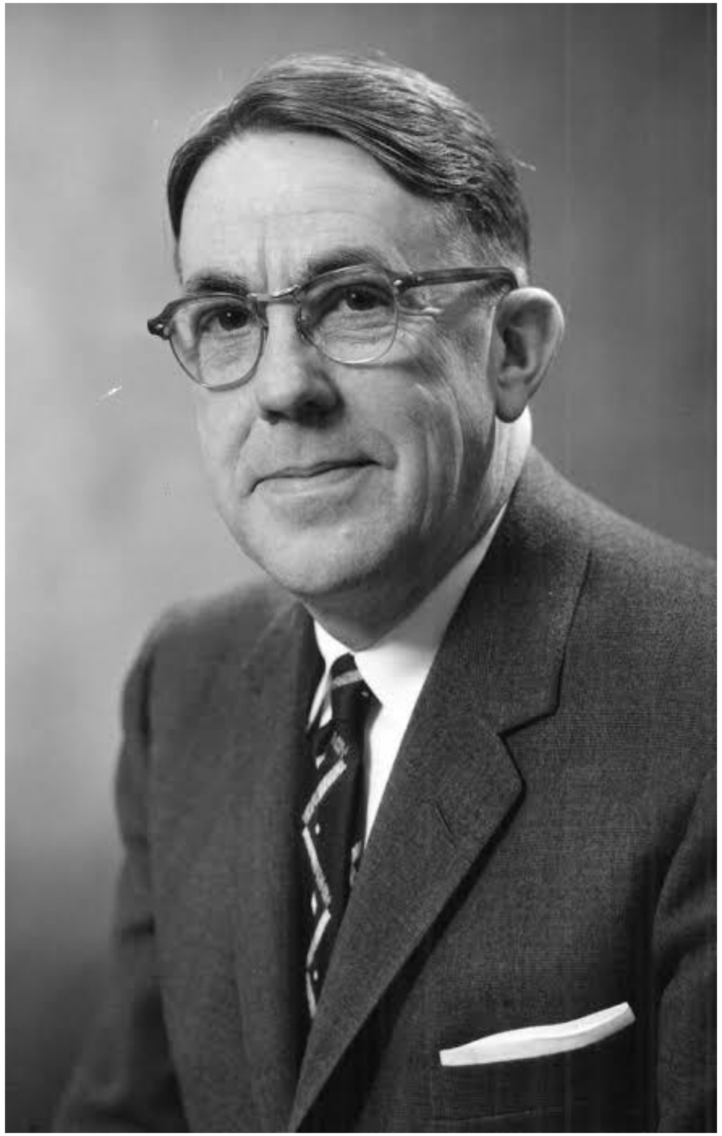
It was a demonstration of the "perceptron" – "the first machine which is capable of having an original idea," according to its creator, Frank Rosenblatt.

The Father of Machine Learning

Arthur Samuel

Arthur Samuel developed the first machine learning program, **Samuel's Checkers**, in **1959**, introducing the concept of machines learning from experience and improving performance over time.





Deep Blue

- IBM's **Deep Blue** used supercomputing power and brute-force computation to defeat world chess champion Garry Kasparov in **1997**, marking a major milestone in AI.

Laying the Foundations of AI

- **Alan Turing (1950):** Proposed the concept of machine intelligence and introduced the Turing Test.
- **Dartmouth Conference (1956):** John McCarthy and others formally established AI as a scientific field.
- **Arthur Samuel (1959):** Arthur Samuel develops a checkers-playing program.
- **1980s-1990s:** Statistical models and neural networks emerge.
- **2000s-Now:** Big Data + Deep Learning → Rapid growth in AI applications.

Milestones in AI Evolution

- **1970s-1980s:** Rise of expert systems, such as MYCIN for medical diagnosis.
- **1997:** IBM Deep Blue defeats world chess champion Garry Kasparov.
- **2012:** AlexNet's success in image recognition competition, sparking the deep learning revolution.
- **2020s:** Generative AI (e.g., ChatGPT) transforms industries with advanced natural language processing.

Generative AI Example

The Famous Image Processing Example of LENA

generate an image of lena flying

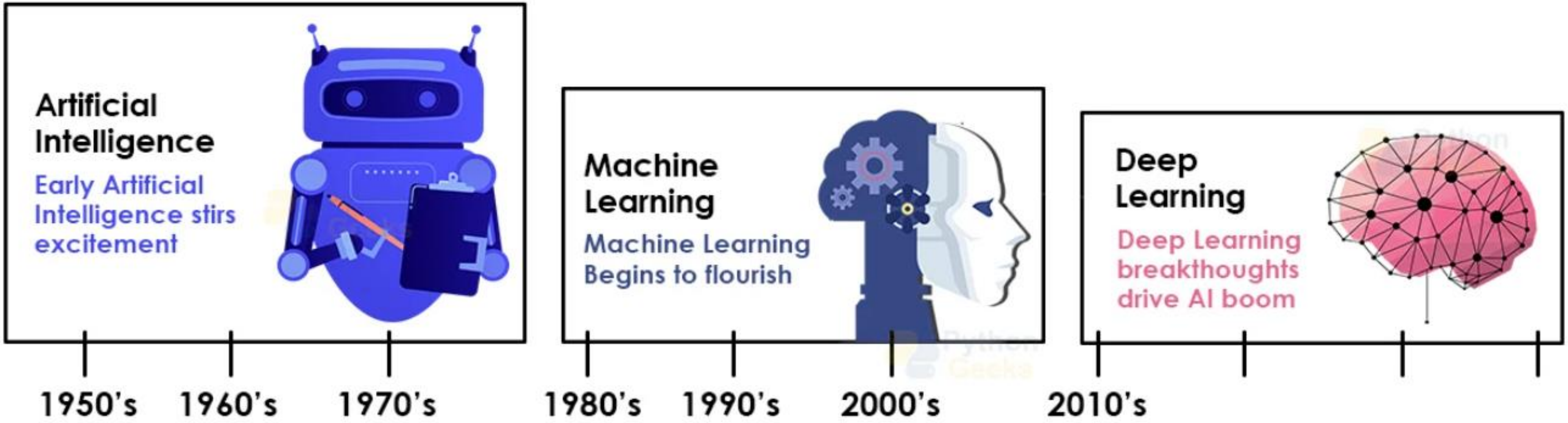
Here is a creative depiction inspired by the famous Lena image, showing her flying gracefully through the sky in a surreal and uplifting setting. Let me know if you'd like further tweaks!

Defining Artificial Intelligence

- **What is AI?**
  - Simulation of human intelligence by machines, including learning, reasoning, and self-correction.
  - **Core Domains:** Computer vision, natural language processing, robotics, and autonomous systems.
- **Philosophical Questions:**
  - Can machines truly "think," or are they only simulating cognition?
  - How do we define consciousness and creativity in machines?
- **Framing AI:**
  - A multidisciplinary field combining computer science, mathematics, psychology, and philosophy.

History of Artificial Intelligence

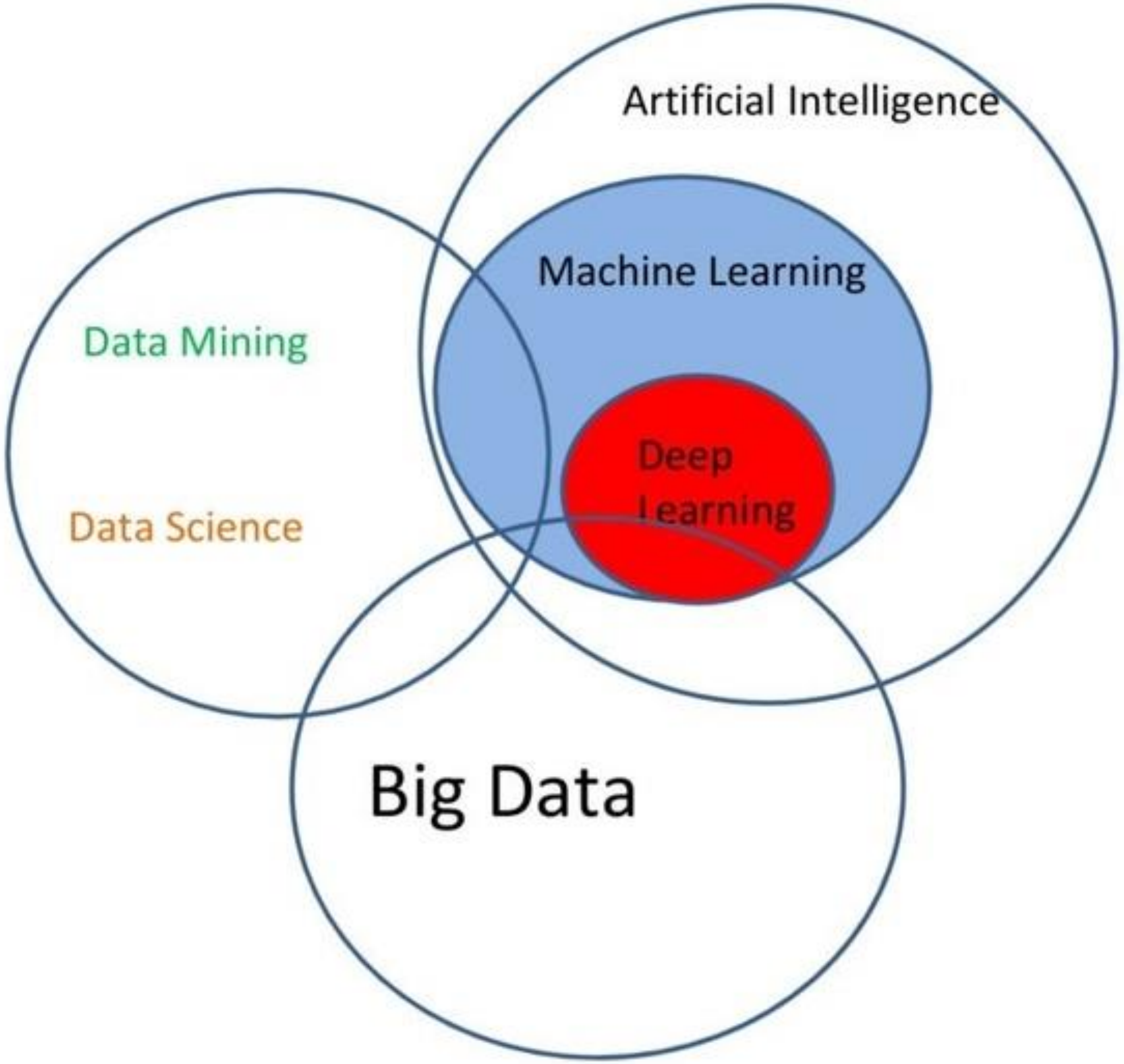
1. **Symbolic AI Era (1950s-1970s):** Rule-based systems dominated. ML experiments were limited by technology.
2. **Statistical Era (1980s-1990s):** Focus shifted to probabilistic models and neural networks.
3. **Modern ML Era (2000s-now):** Big data and advancements in deep learning enabled systems like Siri, GPT, and AlphaGo.



Artificial Intelligence
Early Artificial Intelligence stirs excitement
Machine Learning
Machine Learning Begins to flourish
Deep Learning
Deep Learning breakthoughts drive AI boom
1950's
1960's
1970's
1980's
1990's
2000's
2010's

?

What do you think?



Types of Intelligence

- General vs Narrow AI
- Strong vs Weak AI

Narrow AI (Weak AI)

AI systems designed to perform specific tasks.

Capabilities: Cannot perform tasks outside their predefined purpose.

Examples:

- Virtual assistants like Siri or Alexa.
- Spam email filters.
- Image recognition systems like those used in Facebook photo tagging.
- Recommendation systems (Netflix, Amazon).

Characteristics:

- Task-specific and operates under a limited set of constraints.
- Does not possess consciousness, self-awareness, or general reasoning ability.

General AI (Strong AI)

AI with the ability to perform any intellectual task that a human can do.

Capabilities:

- Learn, understand, and apply knowledge across various domains without additional programming.
- Adaptable and capable of reasoning in unfamiliar situations.

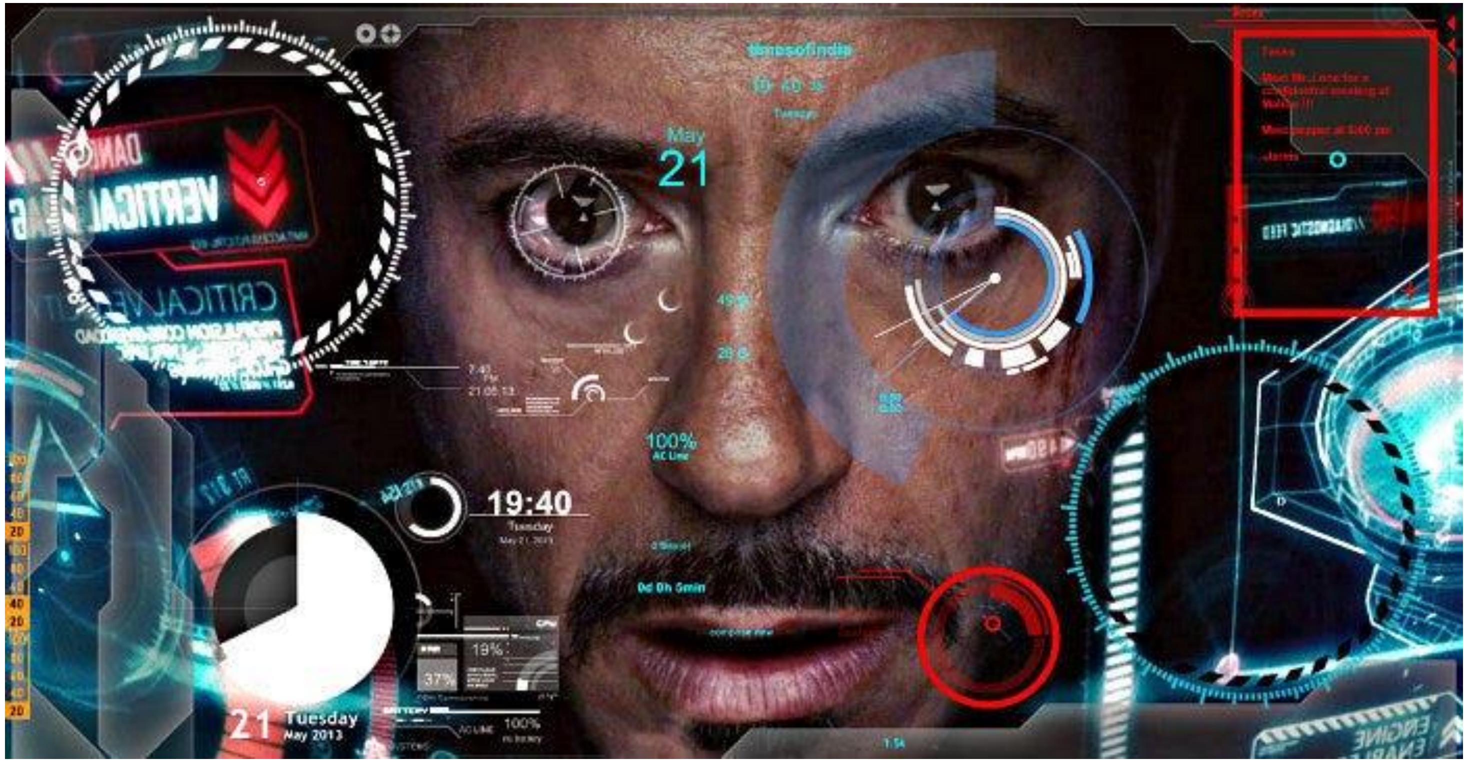
Examples:

- Currently hypothetical, but often represented in fiction (e.g., Jarvis in *Iron Man* or HAL 9000 in *2001: A Space Odyssey*).

Characteristics:

- Can autonomously improve its performance.
- Possesses the ability to reason, plan, solve complex problems, and understand emotions.

General AI (Strong AI) Examples



Jarvis in *Iron Man*

HAL 9000

Difference Between Human and Machine Intelligence

Aspect	Human Intelligence	Machine Intelligence
Definition	Natural cognitive ability to think, learn, and adapt.	Artificially created ability to perform tasks.
Emotions	Capable of emotions and empathy.	Cannot experience emotions.
Learning	Learns through experience, intuition, and reasoning.	Learns through data and algorithms.
Creativity	Highly creative, capable of imagination.	Limited to predefined models and data.
Speed	Relatively slower in processing data.	Extremely fast in processing large datasets.
Adaptability	Adapts naturally to dynamic environments.	Adaptability depends on programming.
Examples	A doctor diagnosing diseases intuitively.	AI analyzing medical images for diagnosis.

Applications of AI

AI has permeated numerous industries and daily life. Key applications include

- **Healthcare**
- **Finance**
- **Education**
- **Transportation**
- **Retail**
- **Manufacturing**
- **Entertainment**
- **Agriculture**
- **Environment**
- **Security**

Healthcare

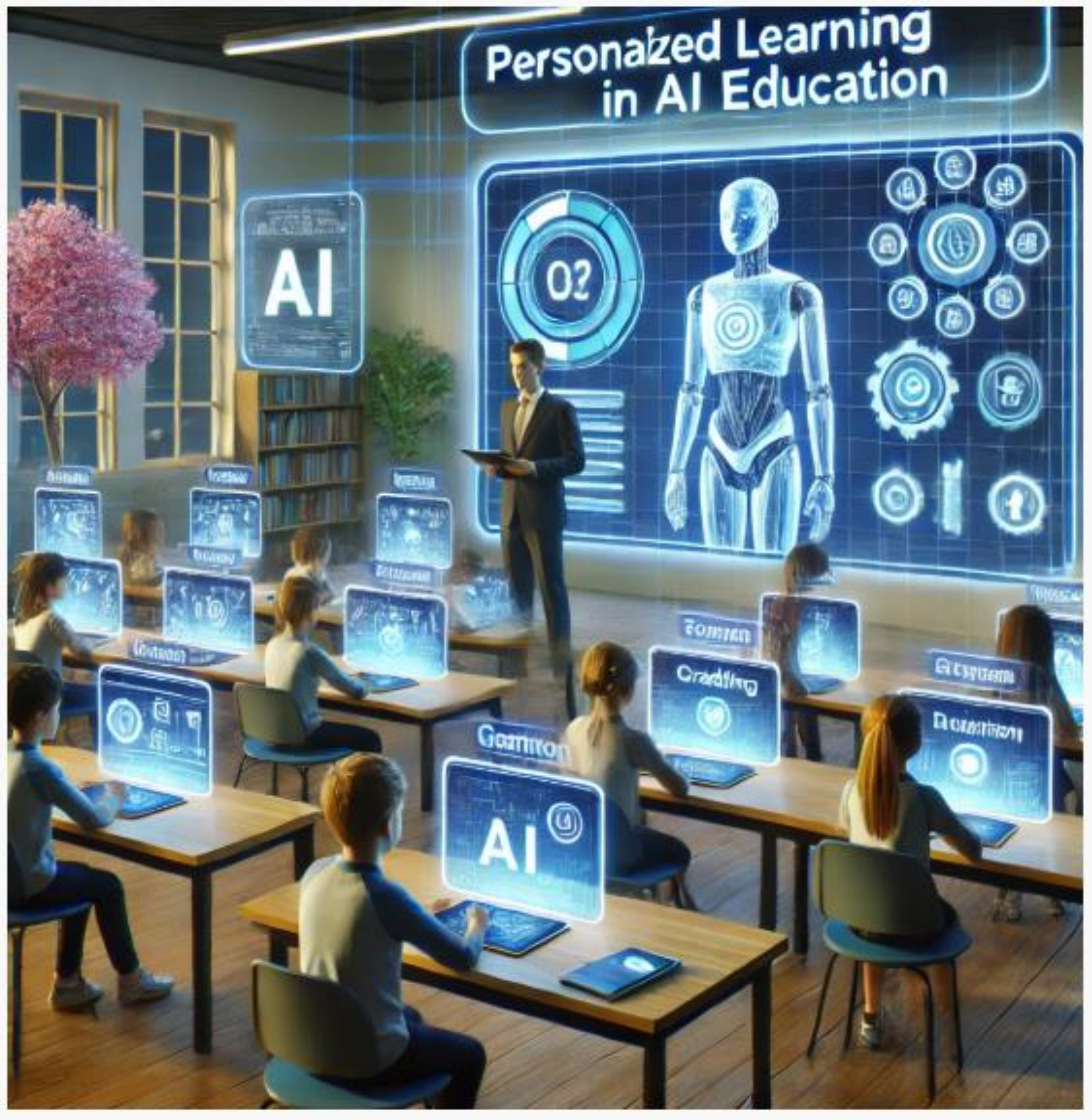
- AI-powered diagnostic tools analyze medical images (e.g., detecting cancer).
- Virtual health assistants provide health advice.
- Predictive analytics for patient care and resource allocation.

Finance

- Fraud detection systems monitor unusual transactions.
- Algorithmic trading optimizes investment strategies.
- Chatbots assist customers with banking queries.

Education

- Personalized learning platforms adapt to students' needs.
- Automated grading systems evaluate assignments efficiently.
- AI tutors provide on-demand learning support.



Transportation

- Autonomous vehicles like self-driving cars.
- AI systems optimize traffic management.
- Predictive maintenance for vehicles and infrastructure.

Retail

- Recommendation engines suggest products to customers.
- Inventory management systems use AI for demand forecasting.
- Chatbots handle customer service inquiries.

Manufacturing

- Robotics streamline assembly lines.
- Predictive maintenance minimizes equipment downtime.
- Quality control systems detect defects in real-time.

Entertainment

- AI curates personalized playlists and content recommendations.
- Virtual reality and augmented reality experiences.
- AI-generated content, such as music or art.

Agriculture

- AI-powered drones monitor crop health.
- Precision farming optimizes resource use.
- Automated harvesting systems.

Environment

- AI monitors and predicts environmental changes.
- Supports wildlife conservation through pattern recognition.
- Optimizes renewable energy production.

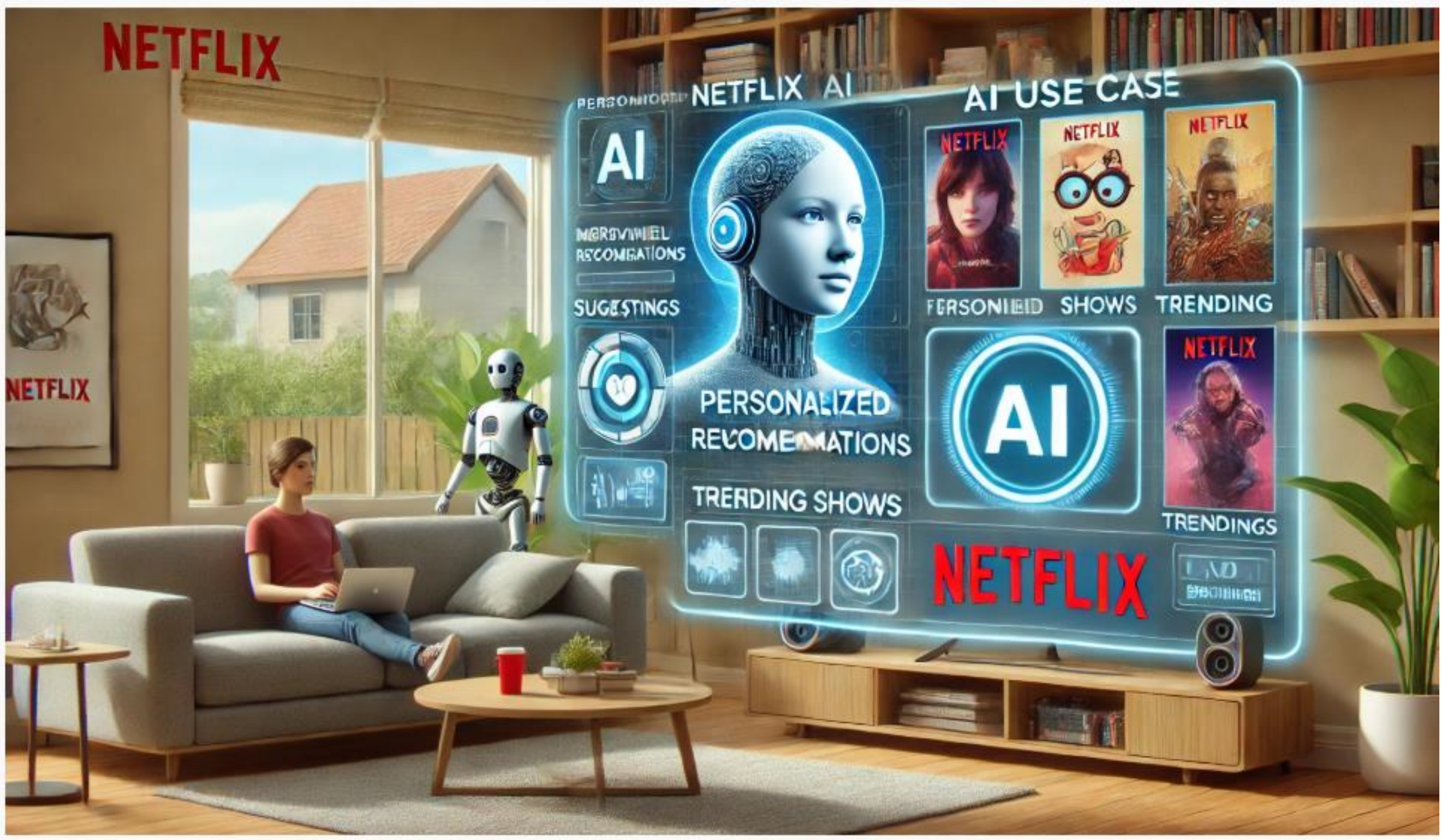
Security

- Facial recognition systems enhance surveillance.
- Cybersecurity tools identify and neutralize threats.
- Disaster response systems analyze and predict natural disasters.

Use Cases of Artificial Intelligence

1. **Netflix's AI-Powered Recommendations** - Personalized movie and show recommendations using AI algorithms.
2. **Tesla's Self-Driving Technology** - Autonomous navigation with AI-powered decision-making.
3. **Amazon's AI-Driven Warehouse Automation** - Robotics and real-time inventory management.
4. **Google's Search Engine Optimization with AI** - User intent analysis and predictive text suggestions.
5. **Facebook's AI Content Moderation System** - Monitoring and flagging policy violations in posts.

Netflix's AI-Powered Recommendations



Tesla's Self-Driving Technology

Amazon's AI-Driven Warehouse Automation

Google's Search Engine Optimization with AI

Facebook's AI Content Moderation System

Future-Focused Applications of AI

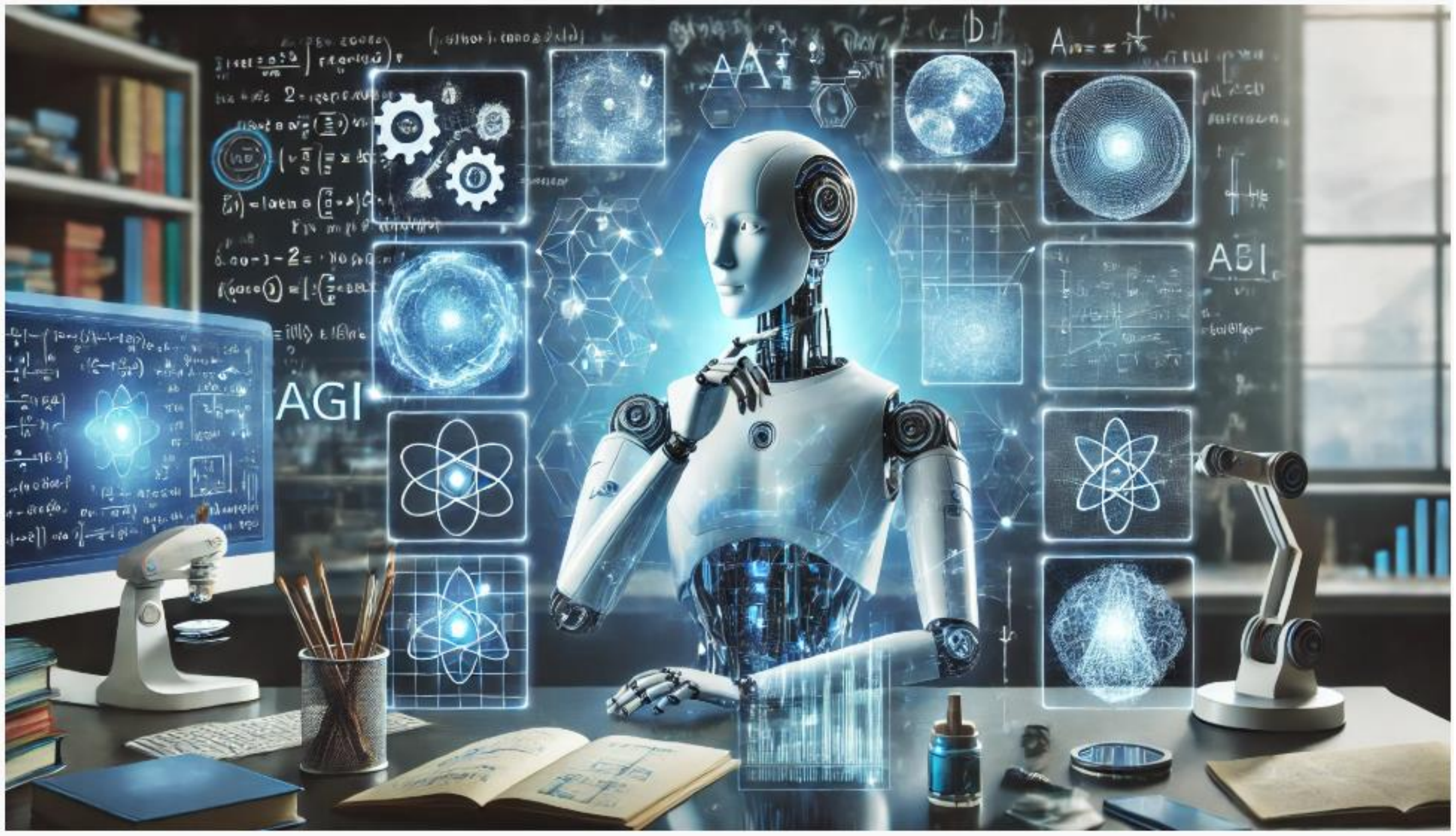
- **Autonomous Vehicles:**
  - Self-driving cars powered by reinforcement learning.
- **Smart Cities:**
  - Optimizing traffic, energy use, and waste management through AI.
- **Artificial General Intelligence (AGI):**
  - Pushing towards machines that can perform any intellectual task a human can.

AI is not just transforming industries; it's shaping the future of innovation across various fields.

Autonomous Vehicles

Smart Cities

Artificial General Intelligence (AGI)



Approaches of Artificial Intelligence

There are several approaches to building AI systems:

- **Symbolic AI (Logic-based)**
- **Machine Learning (Data-driven)**
- **Connectionist Approach (Neural Networks)**
- **Evolutionary Algorithms**
- **Hybrid Systems**

Symbolic AI (Logic-based)

- Relies on predefined rules and symbols to represent knowledge.
- Examples:
  - Expert systems using "if-then" rules.
  - Logic-based reasoning systems.
- Strengths:
  - Transparent and interpretable decision-making.
- Limitations:
  - Cannot handle uncertainty or learn from data.

Machine Learning (Data-driven)

- Relies on algorithms that learn patterns from data.
- Examples:
  - Neural networks for image recognition.
  - Decision trees for classification tasks.
- Strengths:
  - Adaptive and scalable.
- Limitations:
  - Requires large datasets and computational power.

Connectionist Approach (Neural Networks)

- Mimics the structure of the human brain with artificial neurons.
- Examples:
  - Deep learning models like CNNs and RNNs.
- Strengths:
  - Highly effective for complex tasks like image and speech recognition.
- Limitations:
  - Often considered a "black box," lacking transparency.

Evolutionary Algorithms

- Inspired by biological evolution, such as genetic algorithms.
- Strengths:
  - Efficient for optimization problems.
- Limitations:
  - May require significant computational resources.

Hybrid Systems

- Combines multiple approaches (e.g., symbolic AI with machine learning).
- Example:
  - IBM's Watson uses both rule-based reasoning and data-driven learning.

?

How do these approaches lead to systems capable of perceiving, reasoning, and acting effectively?

We’ve explored how AI works and where it’s applied. But how do AI systems make decisions and act on them effectively?

Intelligent Agent

An **Intelligent Agent** is a system that perceives its environment through sensors and acts upon it using actuators to achieve specific goals.

These agents use AI techniques to make decisions autonomously.

Examples: Virtual assistants, self-driving cars, and industrial robots.

Characteristics of Intelligent Agents

- **Autonomy:** Operates without human intervention.
- **Reactivity:** Responds to changes in the environment.
- **Proactiveness:** Takes initiative to achieve goals.
- **Social Ability:** Interacts with other agents or humans.

Examples of Intelligent Agents

- Chatbots like Alexa or Siri.
- Autonomous robots like Roombas.
- Recommendation systems like Netflix or Spotify.

To understand intelligent agents better, let's explore how they are categorized based on their design and functionality.

Types of Agent Programs

- **Simple Reflex Agents**
- **Model-Based Reflex Agents**
- **Goal-Based Agents**
- **Utility-Based Agents**
- **Learning Agents**

Simple Reflex Agents

- Respond directly to percepts using condition-action rules.
- **Example:** A thermostat adjusting temperature based on the current reading.

Simple Reflex Agents

- These agents operate based on *condition-action rules* (if-then rules).
- They observe the current percept and choose an action directly without maintaining any memory of past states.
- **Example:** A thermostat reacts to the current temperature without knowing how the temperature has changed over time.
- **Advantages:**
    - Simple and fast to implement.
- **Limitations:**
    - Cannot handle environments where decisions depend on history or broader context.
    - Fail in partially observable environments.

Model-Based Reflex Agents

- Maintain an internal state to track changes in the environment.
- **Example:** A self-driving car using a map to determine its route.

Model-Based Agents

- These agents maintain an *internal state* to track parts of the world that are not directly observable in the current percept.
- The internal state is updated based on:
  - The agent's perception of the environment.
  - The model of how the world evolves over time.
- **Example:** A robot vacuum cleaner that remembers where it has already cleaned.
- **Advantages:**
  - Better suited for partially observable environments.
- **Limitations:**
  - Complexity increases with the size of the internal state.
  - Relies on accurate models of how the world changes.

Goal-Based Agents

- Act to achieve specific goals by evaluating different possibilities.
- **Example:** A chess program planning moves to win the game.

Goal-Based Agents

- These agents use goals to decide their actions, aiming to achieve specific objectives.
- They use search and planning techniques to determine the sequence of actions needed to achieve their goals.
- **Example:** A chess-playing agent that plans moves to checkmate the opponent.
- **Advantages:**
  - More flexible and capable of adapting to changes in the environment.
- **Limitations:**
  - Computationally expensive due to planning and search.

Utility-Based Agents

- Choose actions that maximize their utility or happiness.
- **Example:** A trading algorithm selecting stocks based on profitability.

Utility-Based Agents

- These agents evaluate the desirability of different states using a *utility function*.
- They aim to maximize their utility, balancing trade-offs among multiple conflicting goals.
- **Example:** An autonomous taxi that minimizes fuel consumption, trip time, and passenger discomfort while maximizing safety and profits.
- **Advantages:**
  - Optimal decision-making when goals conflict.
  - Capable of handling uncertainty and probabilistic environments.
- **Limitations:**
  - Requires a well-defined utility function, which can be challenging to design.

Learning Agents

- Improve performance over time by learning from experiences.
- **Example:** A recommendation system refining suggestions based on user feedback.

Learning Agents

These agents learn from their environment to improve their performance over time.

They consist of four main components:

- **Learning Element:** Responsible for learning and adapting.
- **Performance Element:** Executes actions based on learned knowledge.
- **Critic:** Provides feedback on the agent's performance.
- **Problem Generator:** Suggests new actions for exploration.

- **Example:** A recommendation system refining suggestions based on user interactions.
- **Advantages:**
  - Continuously improves performance and adaptability.
  - Can handle dynamic and complex environments.
- **Limitations:**
  - Learning may require significant data and time.
  - Risk of learning incorrect or biased patterns.

Overview of Agent Types

- **Simple Reflex Agents** are efficient but limited to specific tasks with well-defined rules.
- **Reflex Agents with Internal State** can handle partially observable environments by tracking history.
- **Goal-Based Agents** enable flexible decision-making by focusing on achieving goals through planning.
- **Utility-Based Agents** optimize decisions based on a utility function, making them suitable for complex and uncertain environments.
- **Learning Agents** Improve performance over time by learning from experiences

Simple Reflex → Model-Based → Goal-Based → Utility-Based → Learning.

These structures highlight how increasing the complexity of agent design enhances their decision-making capabilities and adaptability.

Applications of Intelligent Agents

The automated taxi driver combines aspects of all these types:

- **Simple Reflex Behavior:**
    - Reacts to immediate conditions like stopping for pedestrians or avoiding collisions.
- **State Tracking:**
    - Tracks its location, speed, and road conditions using sensors and GPS.
- **Goal-Based Actions:**
    - Plans routes to destinations and navigates traffic efficiently.
- **Utility Maximization:**
    - Balances trade-offs between safety, speed, comfort, and profitability.

Review of Agents

Intelligent agents are systems capable of perceiving their environment, processing the information, and taking actions to achieve specific goals. To act effectively, agents must systematically plan their actions, particularly when solving complex problems

Why Do Agents Need Problem Formulation?

- In dynamic or uncertain environments, simply reacting to stimuli (as a reflex agent would) is insufficient.
- Agents must analyze the problem, anticipate possible outcomes, and evaluate the best course of action using systematic strategies.
- **Systematic Problem-Solving**:
  - Formulating the problem in a structured way is the first step. This allows agents to leverage algorithms that optimize decision-making.

Example

- **A Robot Navigating a Maze**
  - o **Environment**: The maze with walls, paths, and an exit.
  - o **Perception**: Sensors to detect walls and open spaces.
  - o **Goal**: Reach the exit.
  - o **Challenge**: How does the robot decide the path?
- **Questions to Consider:**
  - **Decision-Making**:
    - Should the robot always turn right or left?
    - Can it backtrack if it reaches a dead end?
  - **Optimization**:
    - Can the robot guarantee the shortest path to the exit?
    - What if it encounters obstacles or dynamic changes?

Problem Formulation in AI

Problem formulation is the process of defining a problem in a structured and precise manner, enabling the application of AI techniques to find a solution. It includes defining the initial state, goal state, actions, state space, and constraints.

Steps in Problem Formulation

- **Initial State**: Define the starting condition of the problem.
- **Goal State**: Describe the desired end condition.
- **State Space**: Represent all possible states reachable from the initial state.
- **Actions**: Define the set of operations to transition between states.
- **Constraints**: Highlight any restrictions or limitations in solving the problem.

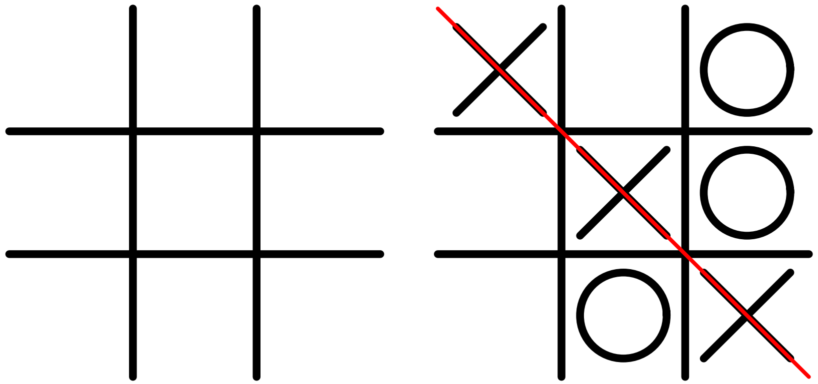
Understanding State Space in Problem Solving

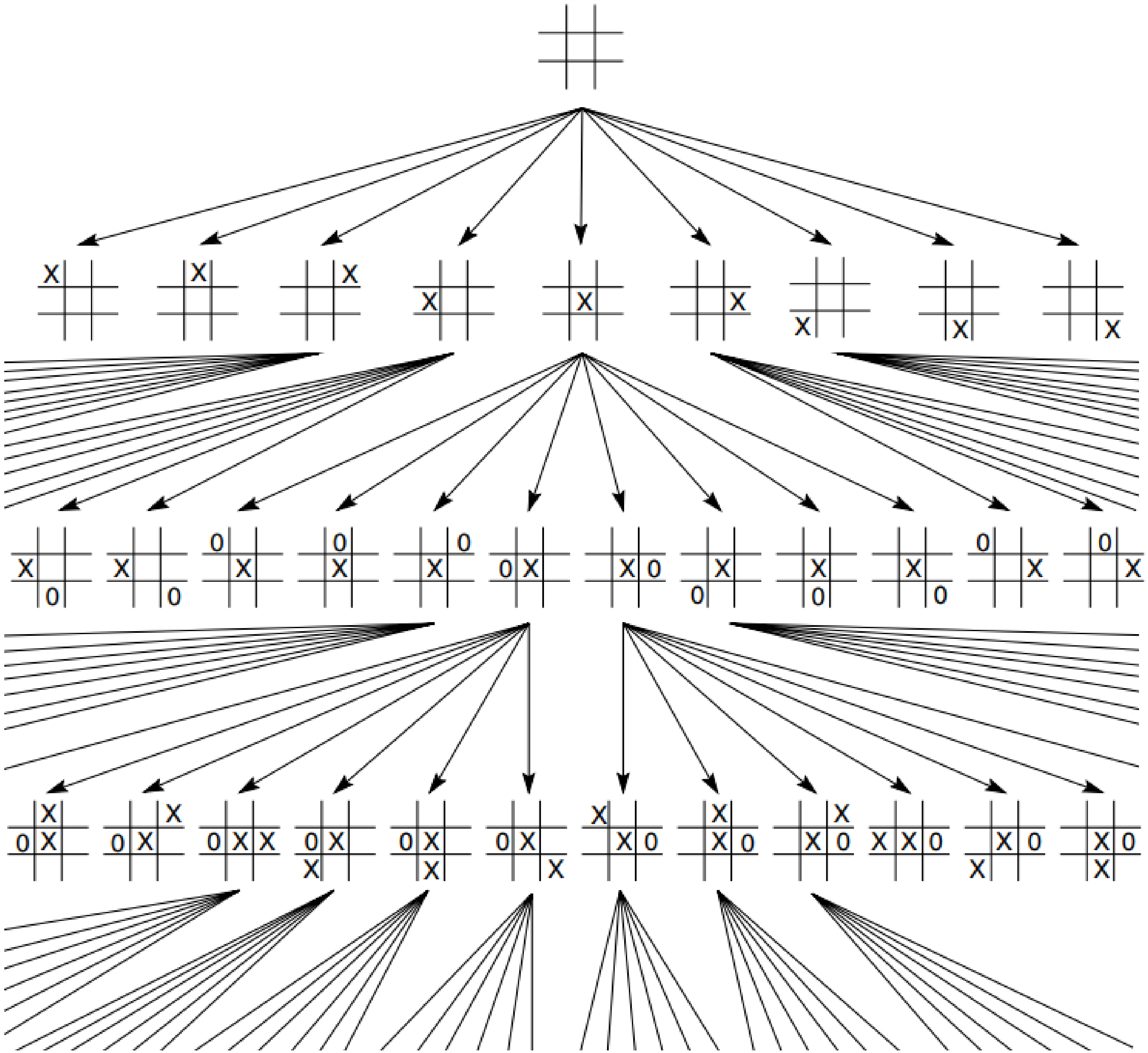
- **Definition:** State space is the complete set of all possible states (configurations) that can be reached from the initial state by applying a series of valid actions.
- **Representation:** A state space is often represented as a graph or a tree, where:
  - o **Nodes:** Represent states.
  - o **Edges:** Represent transitions (actions) between states.
- **Importance:**
  - o Helps in systematically exploring and solving problems.
  - o Essential for defining the scope of intelligent agents' decision-making.

How State Space is Used in Problem Solving

- **Search Algorithms:**
  - Intelligent agents traverse the state space using search techniques like BFS, DFS, or heuristic-based methods.
- **Evaluating Feasibility:**
  - The constraints defined in the problem help prune invalid or redundant states.
- **Efficiency:**
  - State space guides agents to find optimal or near-optimal solutions.

TIC TAC TOE State Space





Examples of Problem Formulation

1. **Water Jug Problem**
2. **8 Puzzle Problem**
3. **Missionaries and Cannibals Problem**

8 Puzzle Problem

1	2	3	4
12	13	14	5
11		15	6
10	9	8	7

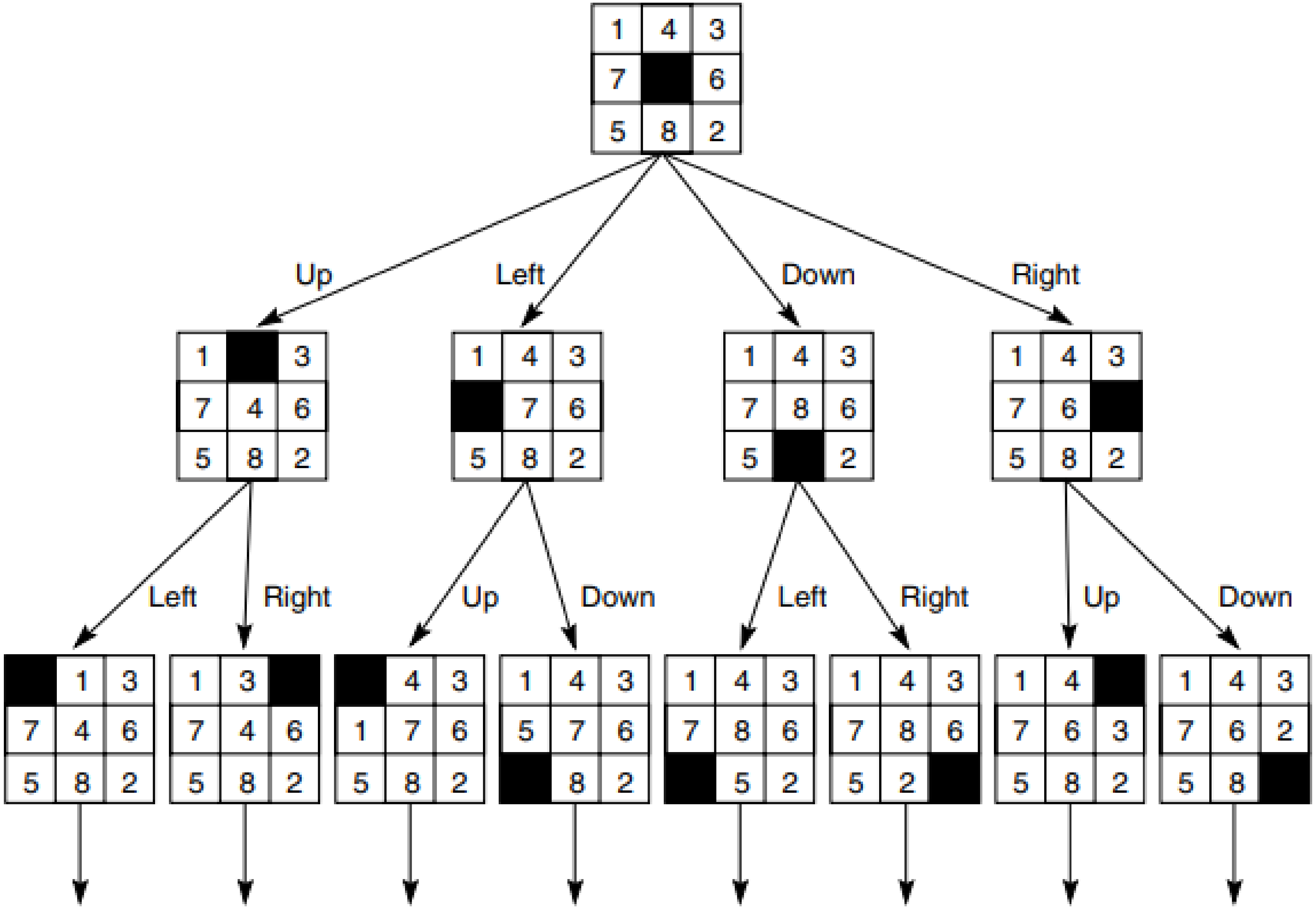
15-puzzle

1	2	3
8		4
7	6	5

8-puzzle

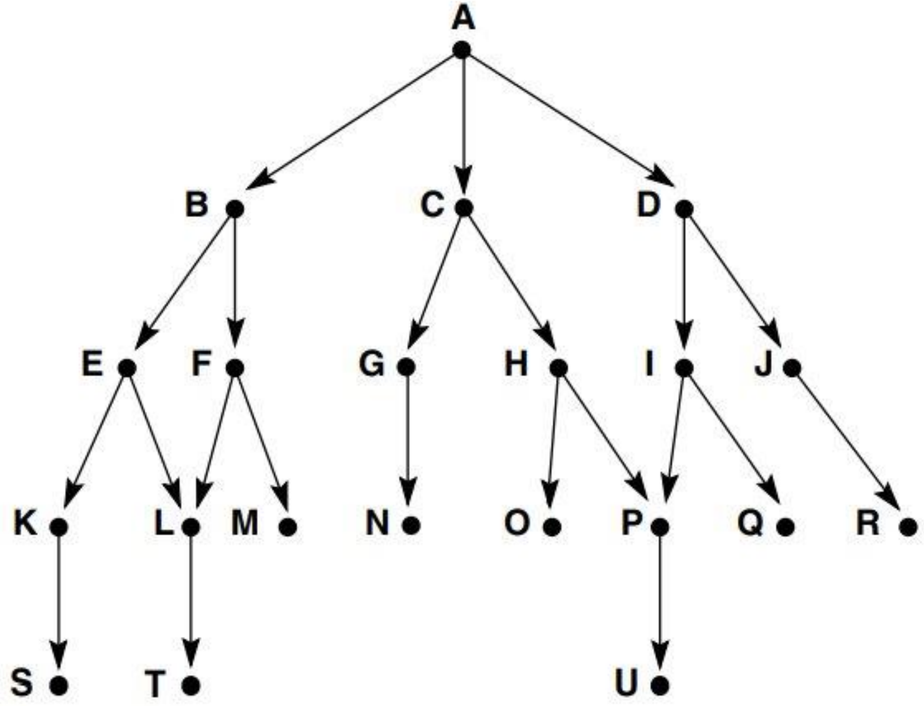
8 Puzzle Problem

- A 3×3 board with numbered tiles (1–8) and one empty space.
- **Objective:**
  - Arrange tiles to match a final configuration by sliding adjacent tiles into the empty space.
- **Possible moves:** Slide tiles **left**, **right**, **up**, or **down**.



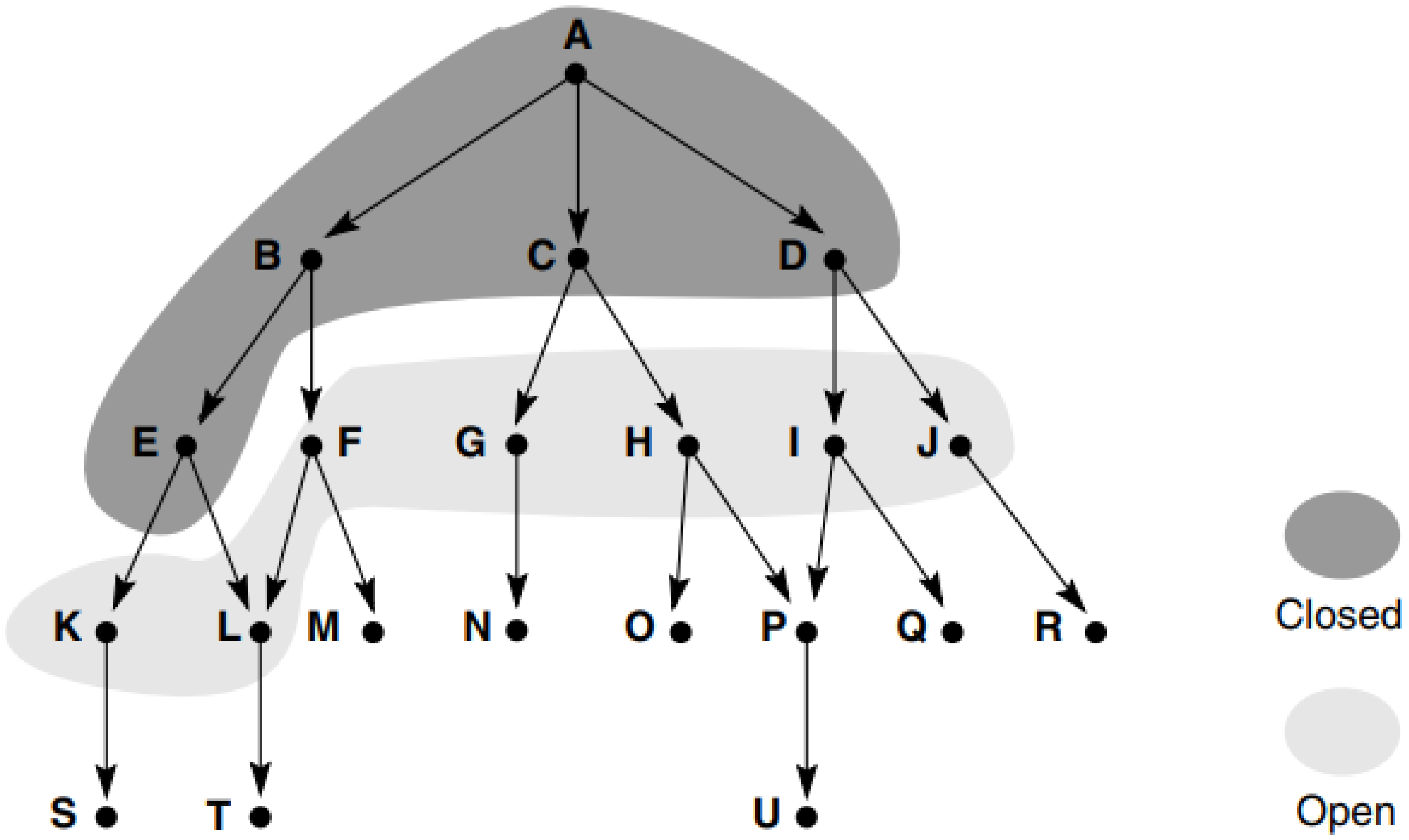
1 4 3
7 6
5 8 2
Up
Left
Down
Right
1 3
7 4 6
5 8 2
1 4 3
7 6
5 8 2
1 4 3
7 8 6
5 2
1 4 3
7 6
5 8 2
Left
Right
Up
Down
Left
Right
Up
Down
1 3
7 4 6
5 8 2
1 3
7 4 6
5 8 2
4 3
1 7 6
5 8 2
1 4 3
5 7 6
8 2
1 4 3
7 8 6
5 2
1 4 3
7 8 6
5 2
1 4
7 6 3
5 8 2
1 4 3
7 6 2
5 8

Breadth First Search



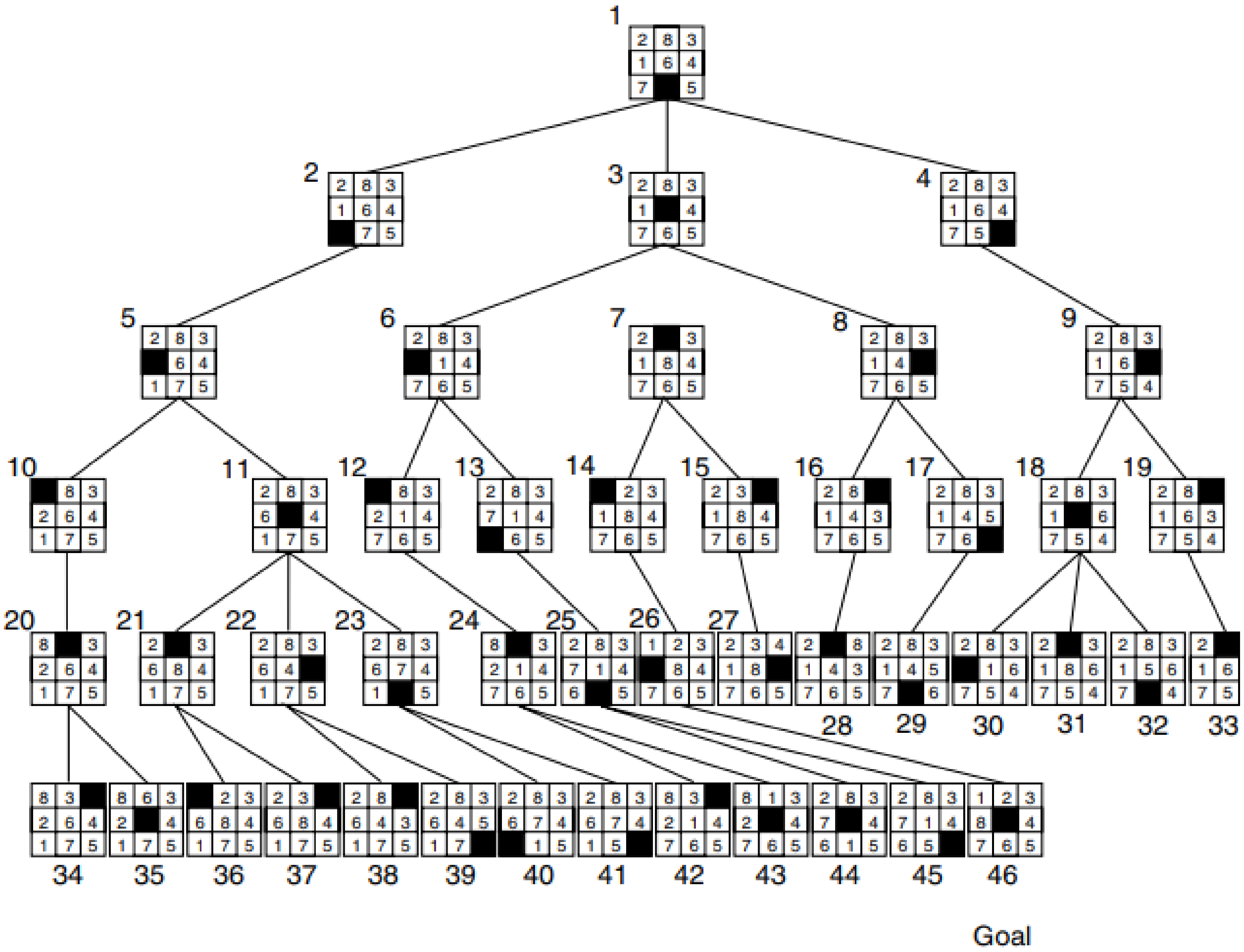
1. open = [A]; closed = [ ]
2. open = [B,C,D]; closed = [A]
3. open = [C,D,E,F]; closed = [B,A]
4. open = [D,E,F,G,H]; closed = [C,B,A]
5. open = [E,F,G,H,I,J]; closed = [D,C,B,A]
6. open = [F,G,H,I,J,K,L]; closed = [E,D,C,B,A]
7. open = [G,H,I,J,K,L,M] (as L is already on open); closed = [F,E,D,C,B,A]
8. open = [H,I,J,K,L,M,N]; closed = [G,F,E,D,C,B,A]
9. and so on until either U is found or open = [ ].

open = [(D,A), (E,B), (F,B), (G,C), (H,C)]; closed = [(C,A), (B,A), (A,nil)]

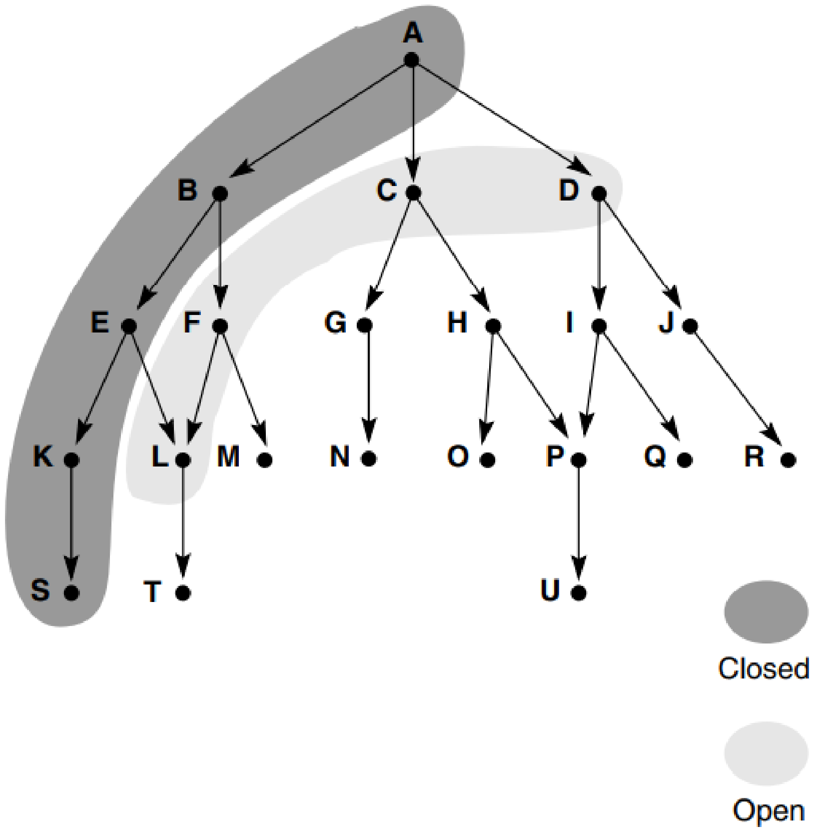


Solving 8-Puzzle with BFS

- Explores all nodes at the current depth before moving to the next level.
- **Approach:**
  - Start at the root node.
  - Explore all neighboring nodes level by level.
  - Continue until the goal state is reached.


Goal

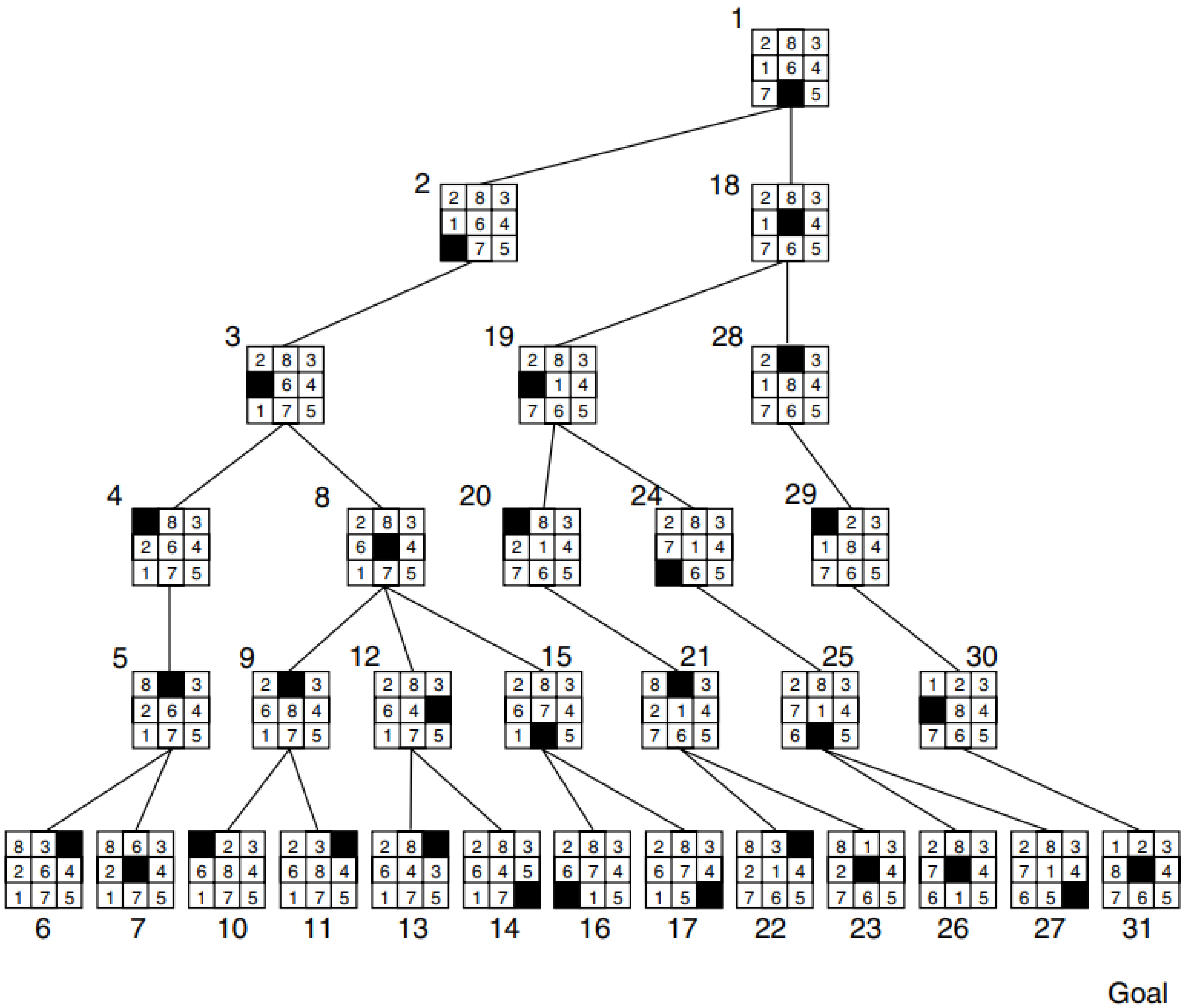
Depth First Search



1. open = [A]; closed = [ ]
2. open = [B,C,D]; closed = [A]
3. open = [E,F,C,D]; closed = [B,A]
4. open = [K,L,F,C,D]; closed = [E,B,A]
5. open = [S,L,F,C,D]; closed = [K,E,B,A]
6. open = [L,F,C,D]; closed = [S,K,E,B,A]
7. open = [T,F,C,D]; closed = [L,S,K,E,B,A]
8. open = [F,C,D]; closed = [T,L,S,K,E,B,A]
9. open = [M,C,D], (as L is already on closed); closed = [F,T,L,S,K,E,B,A]
10. open = [C,D]; closed = [M,F,T,L,S,K,E,B,A]
11. open = [G,H,D]; closed = [C,M,F,T,L,S,K,E,B,A]

Solving 8-Puzzle with DFS

- Explores as far as possible along a branch before backtracking.
- **Approach:**
  - Start at the root node (initial configuration).
  - Explore child nodes recursively.
  - Backtrack when no further moves are possible.


1
2
18
3
19
28
4
8
20
24
29
5
9
12
15
21
25
30
6
7
10
11
13
14
16
17
22
23
26
27
31
Goal

Limitations of DFS and BFS

- **DFS:**
  - Can get stuck in deep paths without progress.
  - High memory usage.
- **BFS:**
  - Explores all paths at the same depth, leading to inefficiency.

- **Solution:**

  Use a smarter algorithm like Branch and Bound.

Water Jug Problem

- The **Water Jug Problem** is a classic puzzle in artificial intelligence and problem-solving, often used to illustrate concepts in state-space representation and search algorithms.
- **Problem Statement:** You are given:
  - Two jugs:
    - Jug **A** with a capacity of mmm liters.
    - Jug **B** with a capacity of n liters, where $0<m<n$.
  - An infinite supply of water.
  - Neither jug has any markings to measure smaller quantities.
- **Goal:** Measure exactly d liters of water ($d<n$) using the two jugs. The problem must be solved using a series of valid operations.

Water Jug Problem

- **Allowed Operations**
  - **Fill a Jug:** Completely fill either jug with water from the infinite supply.
  - **Empty a Jug:** Pour out all the water from a jug.
  - **Pour Water Between Jugs:** Transfer water from one jug to the other until:
    - The first jug is empty.
    - The second jug is full.

Water Jug Problem

- **Key Constraints**
- *d* liters can only be measured if *gcd(m,n)* divides *d*.
  - *gcd(m,n)* is the greatest common divisor of *m* and *n*.
- The solution involves finding the **minimum number of operations** required to measure *d* liters.

Water Jug Problem

- **Example**
  - **Jugs:** m=3, n=5
  - **Target:** d=4 liters
- **Operations Sequence:**
  - Fill the 5-liter jug.
  - Pour water from the 5-liter jug into the 3-liter jug until the latter is full.
  - Empty the 3-liter jug.
  - Pour the remaining water (2 liters) from the 5-liter jug into the 3-liter jug.
  - Fill the 5-liter jug again.
  - Pour water from the 5-liter jug into the 3-liter jug until the latter is full.
  - The 5-liter jug now contains exactly 4 liters.
- **Minimum Operations:** 7

Missionaries and Cannibals Problem

The **Missionaries and Cannibals Problem** is a classic puzzle that demonstrates state-space representation and search algorithms. It is frequently used in Artificial Intelligence to explain problem-solving strategies, such as depth-first search (DFS), breadth-first search (BFS), and heuristic methods.

Missionaries and Cannibals Problem

- **Problem Statement**
  - Three missionaries and three cannibals need to cross a river using a boat.
  - The boat:
    - Can carry at most two people at a time.
    - Cannot cross the river by itself (requires at least one person to row).
    - The goal is to get all six individuals (missionaries and cannibals) safely across the river.
- **Constraints:**
  - On either side of the river, at any point in time, the number of cannibals cannot exceed the number of missionaries. Otherwise, the cannibals will eat the missionaries.
  - The boat can only hold one or two people at a time.

Missionaries and Cannibals Problem

Objective

- Determine a sequence of boat crossings that ensures all three missionaries and three cannibals safely cross the river while following the constraints.
- **State Representation:**
  - Each state can be represented as: *(M_L, C_L, B, M_R, C_R)*
  - Where:
    - M_L and C_L : Number of missionaries and cannibals on the left bank.
    - M_R and C_R : Number of missionaries and cannibals on the right bank.
    - B: Location of the boat ("L" for left bank, "R" for right bank).

Missionaries and Cannibals Problem

- **Initial State:** (3,3,L,0,0)
  - All missionaries and cannibals are on the left bank, and the boat is on the left bank.
- **Goal State:** (0,0,R,3,3)
  - All missionaries and cannibals are on the right bank, and the boat is on the right bank.
- **Valid Transitions:** The boat can carry:
  - One missionary.
  - One cannibal.
  - Two missionaries.
  - Two cannibals.
  - One missionary and one cannibal.
- **Constraints:**
  - At no point should $C_L > M_L$ or $C_R > M_R$, unless $M_L = 0$ or $M_R = 0$

Missionaries and Cannibals Problem

Solution Approach (State Space Search)

- **Generate States:**
  - Start from the initial state.
  - Generate all possible valid moves (boat crossings) that do not violate constraints.
  - Avoid revisiting already explored states.
- **Search Algorithm:**
  - Use **Breadth-First Search (BFS)** for finding the shortest solution.
  - Alternatively, use **Depth-First Search (DFS)** for exploring deeper paths.
- **Path to Goal:**
  - Keep track of the sequence of transitions (state changes) that lead from the initial state to the goal state.

Missionaries and Cannibals Problem

Example Solution

Initial State: (3,3,L,0,0)

Move 1: Boat takes 1 missionary and 1 cannibal to the right. (2,2,R,1,1)

Move 2: Boat returns with 1 cannibal. (2,3,L,1,0)

Move 3: Boat takes 2 missionaries to the right. (0,3,R,3,0)

Move 4: Boat returns with 1 missionary. (1,3,L,2,0)

Move 5: Boat takes 1 missionary and 1 cannibal to the right. (0,2,R,3,1)

Move 6: Boat returns with 1 cannibal. (0,3,L,3,0)

Move 7: Boat takes 2 cannibals to the right. (0,1,R,3,2)

Move 8: Boat returns with 1 cannibal. (0,2,L,3,1)

Move 9: Boat takes 2 cannibals to the right. (0,0,R,3,3)

Solution Length: 9 moves.

Missionary is denoted by 'M' and Cannibal, by 'C'

Rule 1 :	(0, M)	:	One missionary sailing the boat from bank-1 to bank-2
Rule 2 :	(M, 0)	:	One missionary sailing the boat from bank-2 to bank-1
Rule 3 :	(M, M)	:	Two missionaries sailing the boat from bank-1 to bank-2
Rule 4 :	(M, M)	:	Two missionaries sailing the boat from bank-2 to bank-1
Rule 5 :	(M, C)	:	One missionary and one Cannibal sailing the boat from bank-1 to bank-2
Rule 6 :	(C, M)	:	One missionary and one Cannibal sailing the boat from bank-2 to bank-1
Rule 7 :	(C, C)	:	Two Cannibals sailing the boat from bank-1 to bank-2
Rule 8 :	(C, C)	:	Two Cannibals sailing the boat from bank-2 to bank-1
Rule 9 :	(0, C)	:	One Cannibal sailing the boat from bank-1 to bank-2
Rule 10 :	(C, 0)	:	One Cannibal sailing the boat from bank-2 to bank-1

Rules applied and their sequence in Missionaries and Cannibals problem

After application of rule	persons in the river bank-1	persons in the river bank-2	boat position
Start state	M, M, M, C, C, C	0	bank-1
5	M, M, C, C	M, C	bank-2
2	M, M, C, C, M	C	bank-1
7	M, M, M	C, C, C	bank-2
10	M, M, M, C	C, C	bank-1
3	M, C	C, C, M, M	bank-2
6	M, C, C, M	C, M	bank-1
3	C, C	C, M, M, M	bank-2
10	C, C, C	M, M, M	bank-1
7	C	M, M, M, C, C	bank-2
10	C, C	M, M, M, C	bank-1
7	0	M, M, M, C, C, C	bank-2