Artificial Intelligence

Welcome to a fascinating journey into the world of Artificial Intelligence. From learning systems to intelligent automation, AI is the heart of modern innovation. This presentation will help you understand its core principles and real-world uses.

What is AI?

Artificial Intelligence (AI) is the branch of computer science that focuses on **creating machines or software that can perform tasks that normally require human intelligence**.

Main Goals of AI

🧠 Reasoning

Think logically through problems

📚 Learning

Improve from experience and data

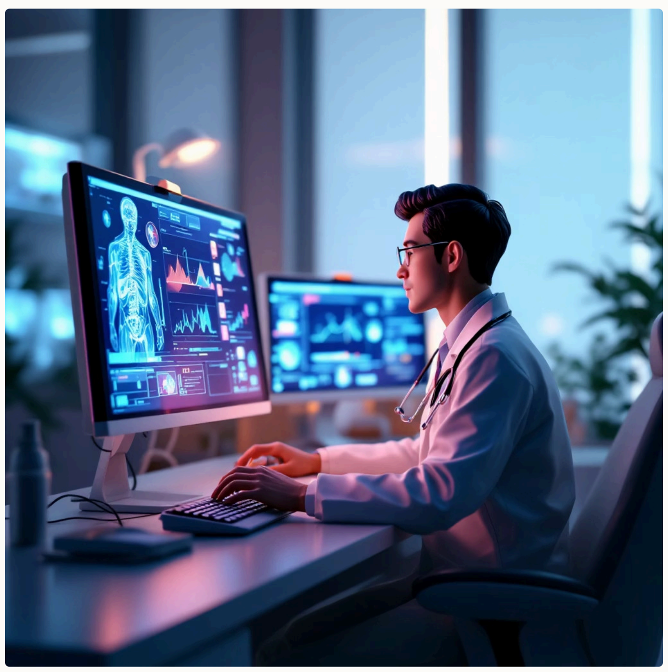
🎯 Problem-Solving

Find optimal solutions efficiently

🤖 Autonomy

Act independently and intelligently

Real-World AI Applications



Healthcare AI

AI assists doctors in diagnosing diseases, analyzing medical scans, and developing personalized treatment plans for patients.

Industrial Robotics

AI-powered robots perform precise manufacturing tasks, quality control, and assembly line operations with high efficiency.

Gaming AI

Intelligent NPCs, adaptive difficulty systems, and procedural content generation create immersive gaming experiences.

State Space Search

State space search is a fundamental concept in artificial intelligence, used for solving problems by exploring possible states to find a path from an initial state to a goal state.

State Space

Initial State

Goal State(s)

Actions & Transitions

Foundations of Intelligence

Neuroscience:

Neuroscience is the **scientific study of the nervous system**, including the brain, spinal cord, and networks of neurons. It focuses on understanding **how the brain processes information, controls behavior, and enables learning and cognition**.

Rationality:

Building systems that make optimal decisions given available information. Logic and decision theory provide rigorous foundations for rational behavior.

The Turing Test

The **Turing Test** is a method proposed by **Alan Turing (1950)** to determine whether a machine can exhibit human-like intelligence.

How it works:

1. A human evaluator communicates with two participants through a computer interface:
   - One is a **human**.
   - The other is a **machine (AI)**.
2. The evaluator asks questions and receives answers from both participants.
3. If the evaluator **cannot reliably tell which is the machine**, the AI is said to have **passed the Turing Test**.

Purpose: To test a machine's ability to **think and respond like a human**, without relying on physical appearance.

Example: Chatbots like **ChatGPT** attempt to mimic human conversation.

Agents In AI:

An agent is an autonomous computational entity that perceives its environment through sensors and acts upon it through actuators. Agents form the conceptual foundation for most AI systems, enabling sophisticated autonomous behavior through perception-action cycles.

Goals Of Agents:

- High performance
- Optimal result
- Rational actions

Types Of Agent:

- Simple reflex agent.
- Model based agent.
- Goal based agent.
- Utility based agent.
- Learning agent.

Types of AI Environments

Fully Observable vs Partially Observable

Fully Observable: Agent has complete information about environment state (Example: Chess - all pieces visible)

Partially Observable: Agent has limited or incomplete information (Example: Self-driving cars in fog)

Deterministic vs Stochastic

Deterministic: Next state completely determined by current state and action (Example: Tic-tac-toe)

Stochastic: Next state involves randomness or uncertainty (Example: Poker with card shuffling)

Static vs Dynamic

Static: Environment doesn't change while agent is thinking (Example: Crossword puzzle)

Dynamic: Environment changes while agent decides (Example: Real-time traffic system)

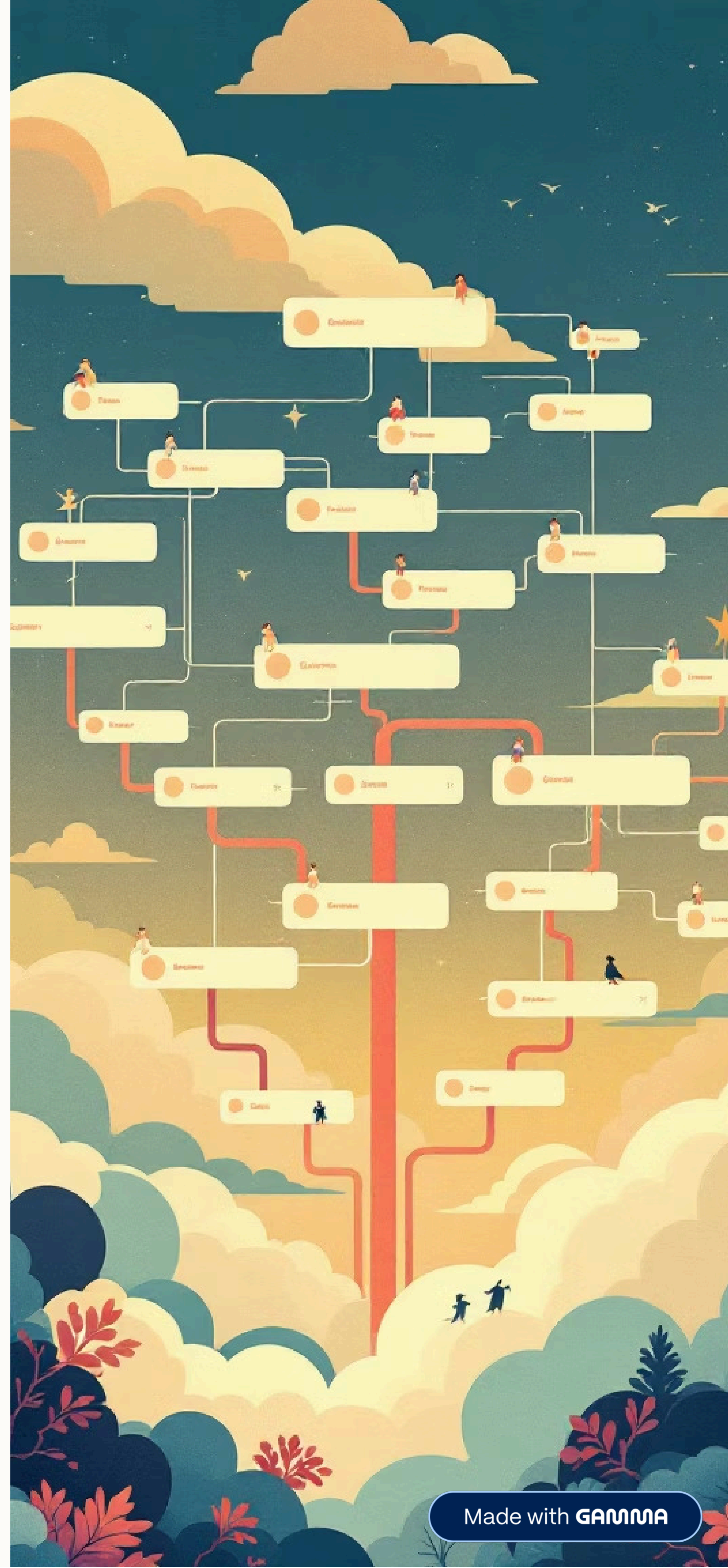
Discrete vs Continuous

Discrete: Finite number of states and actions (Example: Chess moves)

Continuous: Infinite or real-valued states and actions (Example: Robot arm movement)

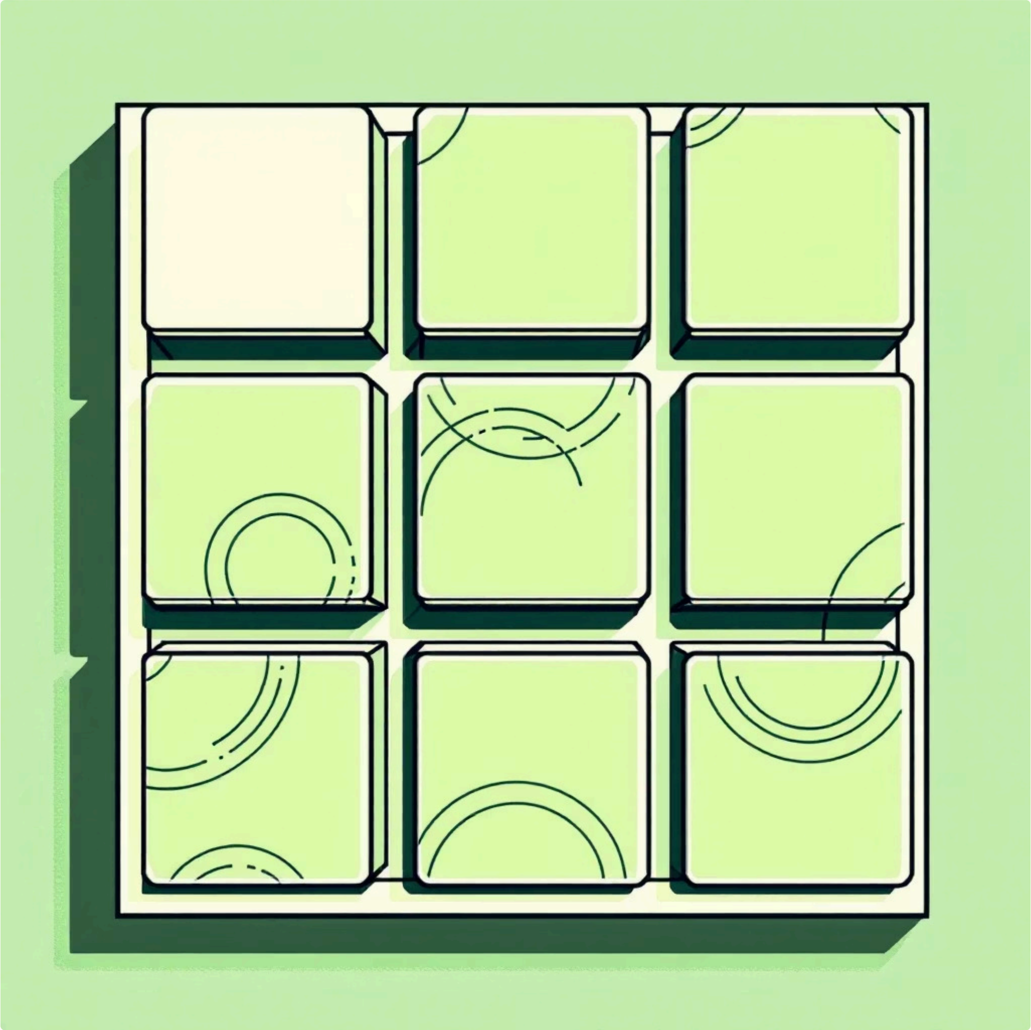
Search Strategies

1	**Uninformed Search** **BFS:** explores all nodes level by level. **DFS:** explores deep into each path. **Bidirectional:** searches from start and goal simultaneously.
2	**Depth-Limited Search** Sets maximum depth limit to prevent infinite exploration
3	**Informed Search** **Greedy Best-First:** uses heuristic. **A*:** combines actual cost with heuristic estimate. **Beam Search:** limits nodes explored.
4	**Local Search Algorithms :** **Hill Climbing:** Iteratively moves towards a state with a better value, continuing until a local optimum is reached. Prone to getting stuck in local maxima.



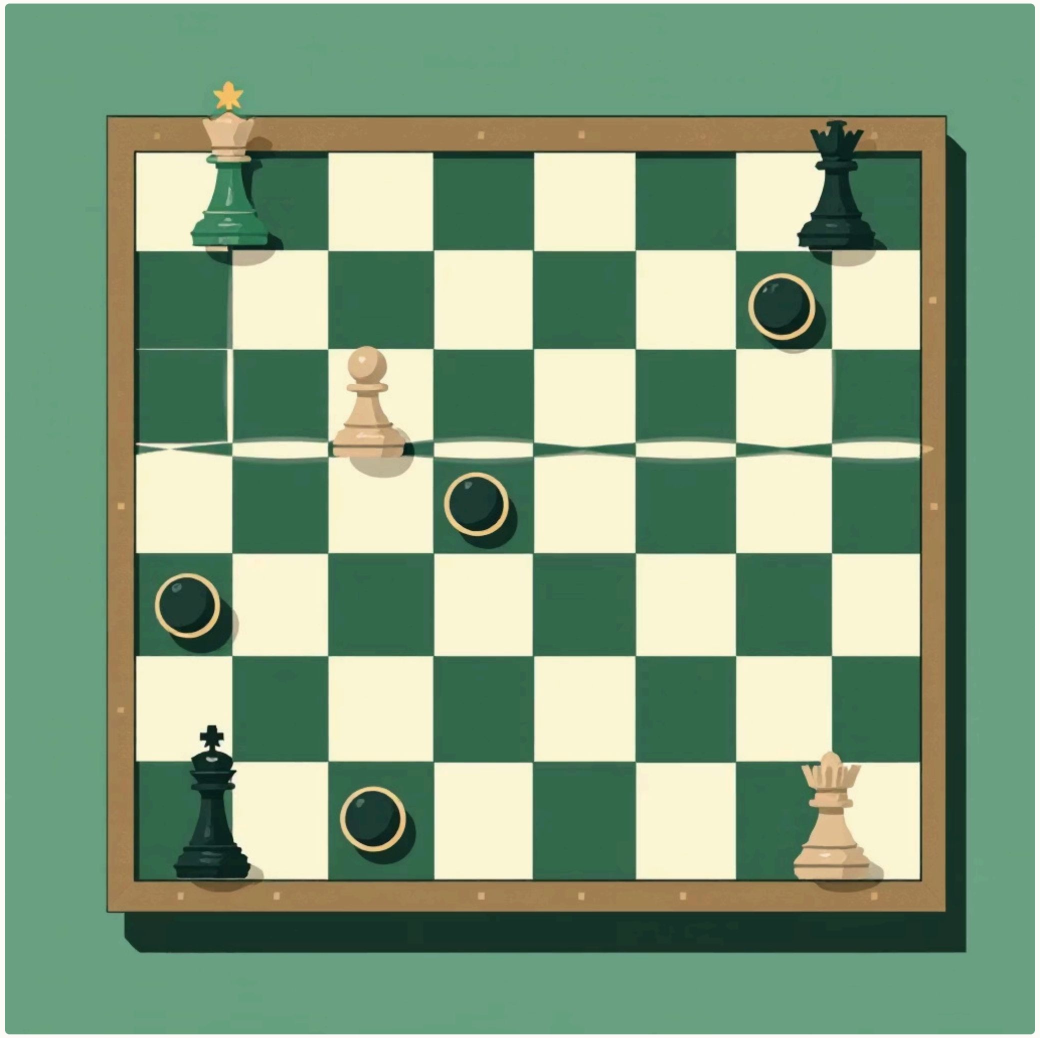
Classic AI Problems

8-Puzzle



Arrange numbered tiles. Test for heuristics and search efficiency.

N-Queens



Place queens without attacks. Tests constraint satisfaction.

Game-Playing Algorithms

Minimax

Maximize player score, minimize opponent's

Alpha-Beta Pruning

Eliminate branches that won't affect outcome

Game AI

Chess, Tic-Tac-Toe, Rock-Paper-Scissors

AI Games :

Tic Tac Toe

Simple perfect-information game. Min-Max solves it completely, leading to draw with optimal play. Demonstrates basic game-tree evaluation.

Chess

Enormous search space (10^120 possible games). Combines Min-Max with alpha-beta pruning, heuristic evaluation, and opening/endgame databases.

Rock Paper Scissors

Simple probabilistic game enhanced with AI using Markov-chain prediction and pattern-based counter moves.

Optimality Conditions

Admissibility

Heuristic never overestimates true cost. Essential for optimality.

Consistency (Monotonicity)

Each step cost plus heuristic respects triangle inequality.

Completeness

Algorithm finds solution if one exists.

Well-Defined Problems

Clear initial state, goal test, operators, and path cost.

Constraints in Artificial Intelligence

Constraints define restrictions or requirements that solutions must satisfy. They are essential for modeling real-world problems and guiding AI systems toward valid, practical answers by reducing the search space and ensuring solutions meet specific domain requirements.

Understanding Constraint Satisfaction Problems (CSPs)

CSPs are a fundamental framework in AI for modeling and solving problems by finding a state that satisfies a given set of constraints. A CSP is formally defined by three components:

1

Variables

These are the unknown elements that need to be assigned values. For example, in a timetabling problem, variables could be the courses that need to be scheduled.

2

Domains

Each variable has a domain, which is a finite set of possible values it can take. For a course scheduling variable, its domain might be a list of available time slots and classrooms.

3

Constraints

These are the conditions that limit the possible values variables can take, or combinations of values for multiple variables. Constraints define what constitutes a valid solution.

Types of Constraints

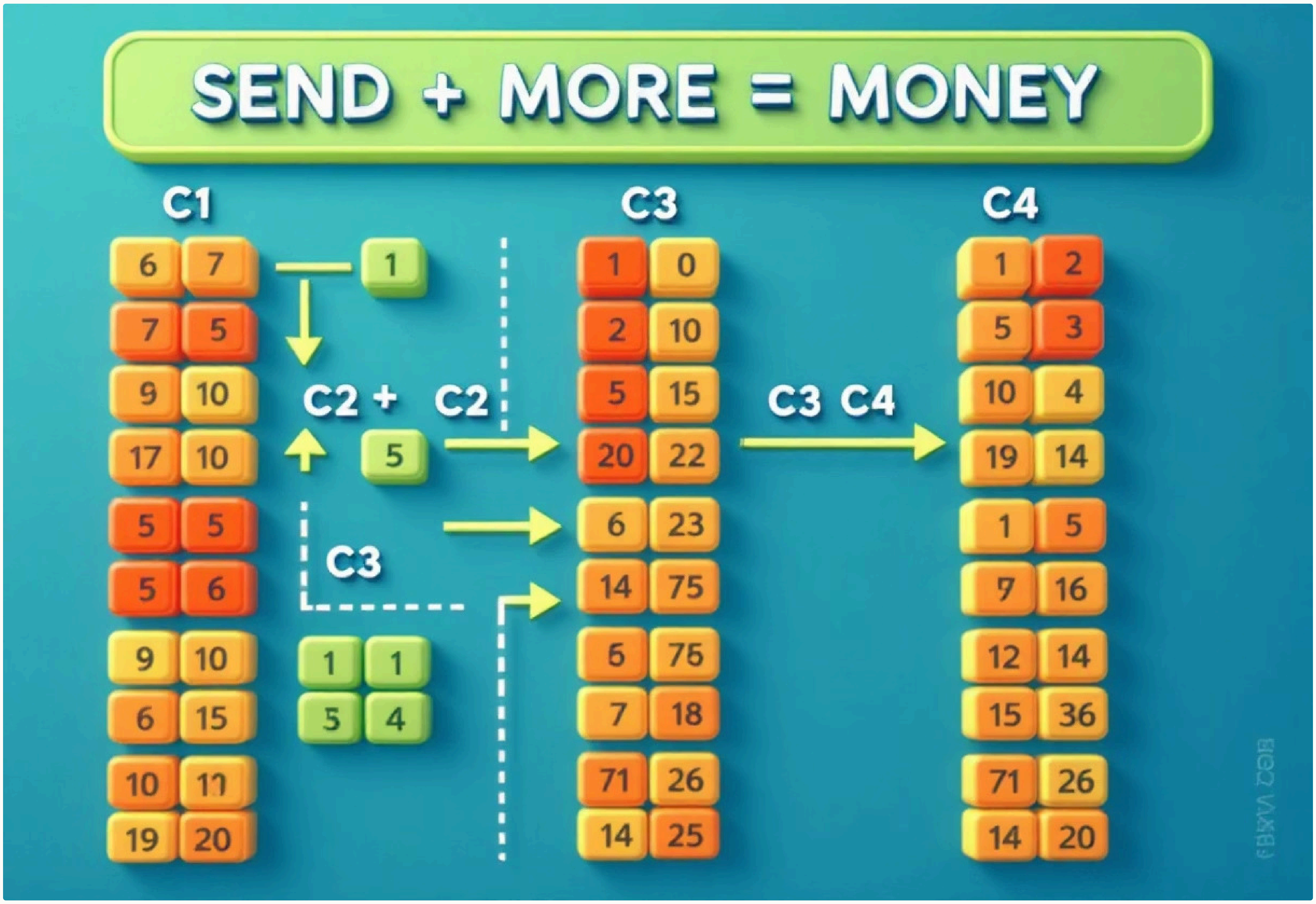
- **Binary constraints**
- Unary constraints
- Global constraints

Cryptarithmetic & Bound Propagation

Cryptarithmetic puzzles are a classic type of constraint satisfaction problem (CSP) where letters represent unique digits (0-9) and mathematical equations must hold true. The goal is to assign a unique digit to each letter such that the arithmetic problem is valid. These puzzles are excellent demonstrations of how AI algorithms use constraints and propagation to efficiently search for solutions.

The Classic Example: SEND + MORE = MONEY

Consider the well-known puzzle:



Here, each letter (S, E, N, D, M, O, R, Y) must be replaced by a unique digit from 0 to 9.

Key Constraints Involved:

- **Unique Digit Mapping:** Each distinct letter must represent a unique digit (0-9). For example, if S=9, no other letter can be 9.
- **Leading Digits:** The leading letters of each number (S and M in SEND and MORE, and M in MONEY) cannot be zero. (S ≠ 0, M ≠ 0)
- **Carry Operations:** The sum involves "carry-overs" (C1, C2, C3, C4) from one column to the next, similar to manual addition. These carries can only be 0 or 1.

Thank You