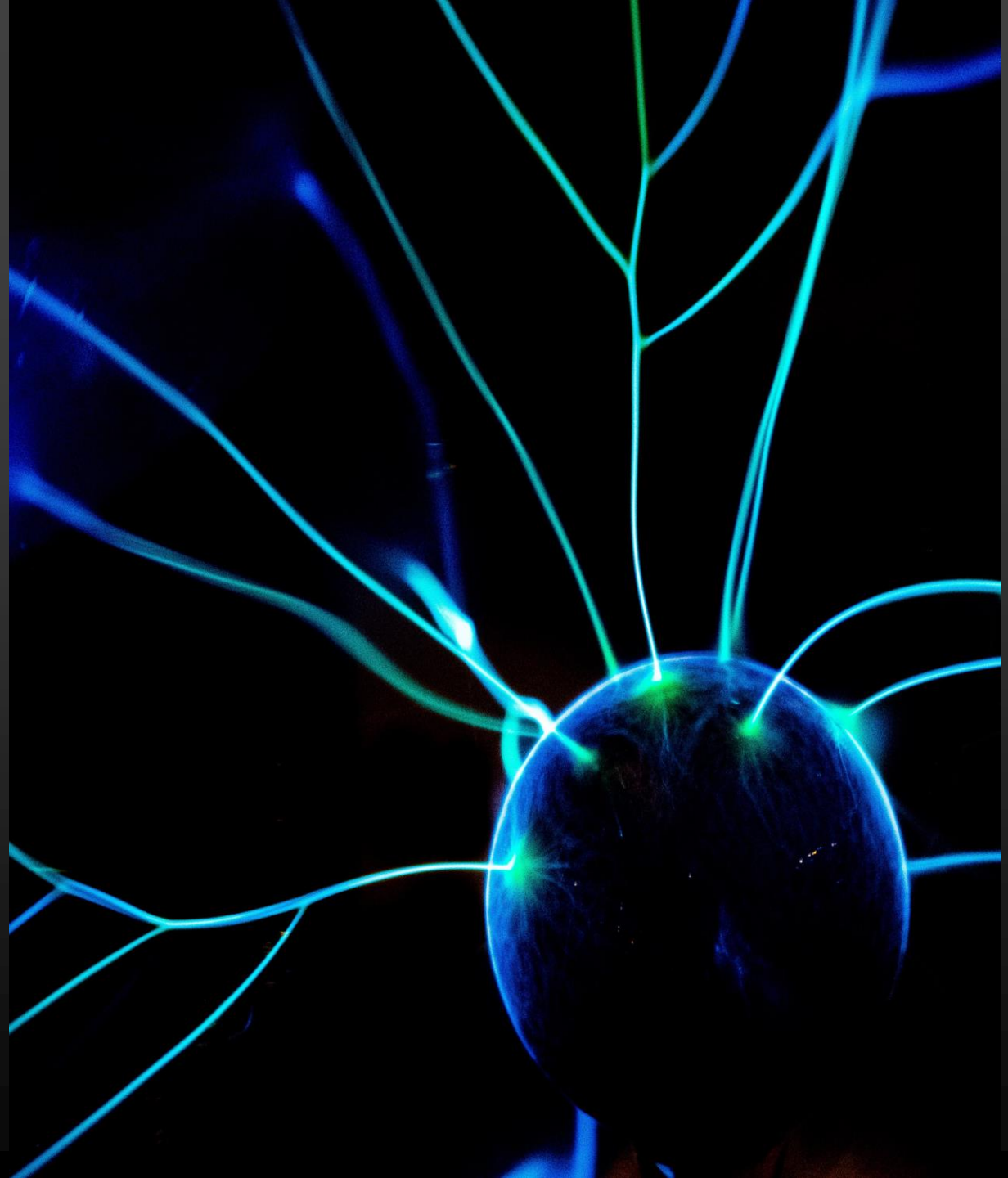
Searching Techniques

With Python

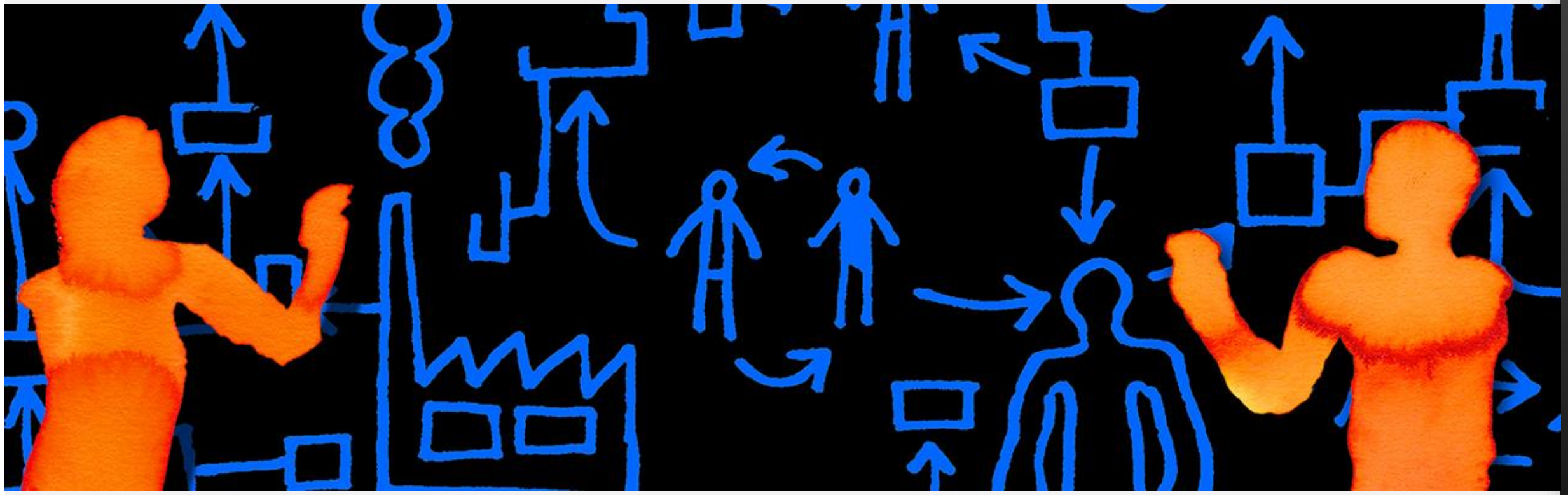


Introduction to Searching Techniques

Search techniques are universal problem-solving methods. **Rational agents** or **Problem-solving agents** in AI mostly used these search strategies or algorithms to solve a specific problem and provide the best result. Problem-solving agents are the goal-based agents and use atomic representation

Searching techniques play a crucial role in computer science and information retrieval, enabling efficient and effective exploration of data to locate specific information. These techniques are used in various applications, ranging from databases and search engines to artificial intelligence and problem-solving algorithms. The goal is to quickly and accurately find the desired information within a dataset, which could be as simple as a list of elements or as complex as a database of documents.

Problem Solving by Searching:



Components of Problem Solving by Searching: ➔

Components of Problem Solving by Searching:

State Space:	*Definition:* The set of all possible configurations or states that the problem can have.
	Example: In the Eight-Puzzle problem, each possible arrangement of the tiles on the board represents a state.
Initial State:	*Definition:* The starting configuration of the problem.
	Example: In the Eight-Puzzle problem, the initial arrangement of tiles on the board.
Goal State:	*Definition:* The desired or target configuration that the problem solver aims to reach.
	Example: In the Eight-Puzzle problem, a specific arrangement of tiles representing the solved state.
Operators (Actions):	*Definition:* Actions or operations that can be applied to move from one state to another.
	Example: In the Eight-Puzzle problem, operators might include moving a tile to an adjacent empty space.
Transition Model:	*Definition:* Describes the result of applying an operator to a particular state.
	Example: The transition model for the Eight-Puzzle defines how the state changes when a tile is moved.
Path Cost:	*Definition:* The cost or distance associated with a path from the initial state to a particular state.
	Example: In a navigation problem, the path cost might represent the distance traveled.
Search Space:	*Definition:* The entire set of states reachable from the initial state by applying sequences of operators.
	Example: In the Eight-Puzzle, the search space includes all possible board configurations.
Search Algorithm:	*Definition:* A systematic procedure for exploring the search space to find a solution.
	Examples: Depth-First Search (DFS), Breadth-First Search (BFS), A* Search.

Problem Solving Strategies:

Uninformed Search Algorithms:

1. *Description:* Search algorithms that explore the search space without using any specific information about the problem.
2. *Examples:* Breadth-First Search, Depth-First Search.

Informed Search Algorithms:

1. *Description:* Search algorithms that use heuristic information to guide the search towards more promising paths.
2. *Examples:* A* Search, Greedy Best-First Search.

Iterative Deepening:

1. *Description:* A hybrid strategy that combines the advantages of depth-first and breadth-first search by gradually increasing the depth of the search.

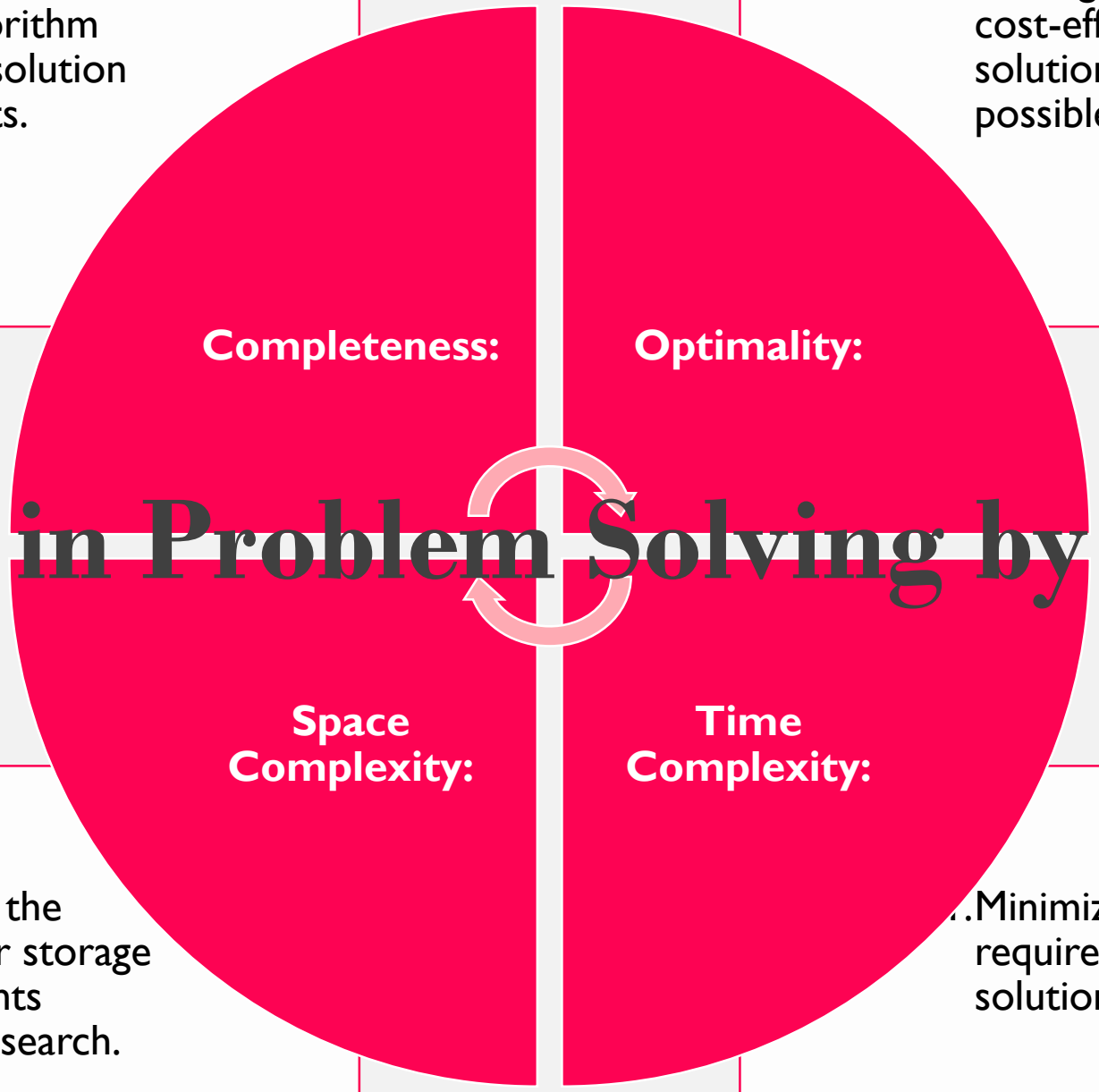
Bidirectional Search:

1. *Description:* Simultaneously performs a forward search from the initial state and a backward search from the goal state, aiming to meet in the middle.

Heuristics:

1. *Definition:* Domain-specific knowledge or rules that estimate the cost or desirability of different states.
2. *Example:* In the Eight-Puzzle, a heuristic might be the number of tiles out of place.

Challenges in Problem Solving by Searching:



1. Ensuring that the search algorithm will find a solution if one exists.

1. Finding the most cost-effective solution among the possible solutions.

1. Minimizing the memory or storage requirements during the search.

1. Minimizing the time required to find a solution.

#kamalnainx

+91 9999 8400 82

@KamaLnainX | @TheKamalNain

Applications:

Route Planning:

Finding the optimal path in transportation or navigation.

Puzzle Solving:

Solving puzzles like the Eight-Puzzle or Sudoku.

Game Playing:

Developing strategies for games like chess or tic-tac-toe.

Artificial Intelligence Planning:

Generating plans and sequences of actions in AI systems.

Optimization Problems:

Finding the optimal solution in various domains, such as resource allocation.

Thank You

Kamalnainx

https://github.com/kamalnainx

+91 9999 8400 82

@KamaLnainX | @TheKamalNain