Problem Solving by Searching

Faculty of Emerging Sciences and Technologies (FEST)

Topics:

- Problem Solving by Search
  - Uninformed Search
  - Informed Search
  - Heuristics
  - Local Search
  - Minimax
  - Alpha–Beta
  - Game-Playing

Problem Solving?

- **Problem solving** = finding sequence of actions from start → goal.
- **Search space** = all possible states.
- **Solution** = path from start to goal.

- Example: A farmer wants to find best irrigation method.
- Analogy: A maze game.

Problem-Solving Agent

- Defines the problem (start & goal).
- Searches through actions.
- Chooses best path.
- Example:
  Student planning career:
  Intermediate → University → Degree → Job.
- Question: What is your goal state?

Problem-Solving Agent

Problem-solving agent = an AI system that **looks ahead**.

Works in steps: **State** → **Action** → **New State.**

All possibilities = **search space**.

Analogy: A student preparing for CSS exam → every choice of subject is a branch in a tree. The agent searches for the best subject combination.

Uninformed Search (Blind)

- **Uninformed search** means the agent has **no extra knowledge** about the goal beyond the problem definition.
- It only knows:
  - Where it starts (start state).
  - What it wants (goal test).
  - What actions are possible.
- It searches **systematically**, but **blindly**, without guidance. That’s why it’s also called **blind search**.

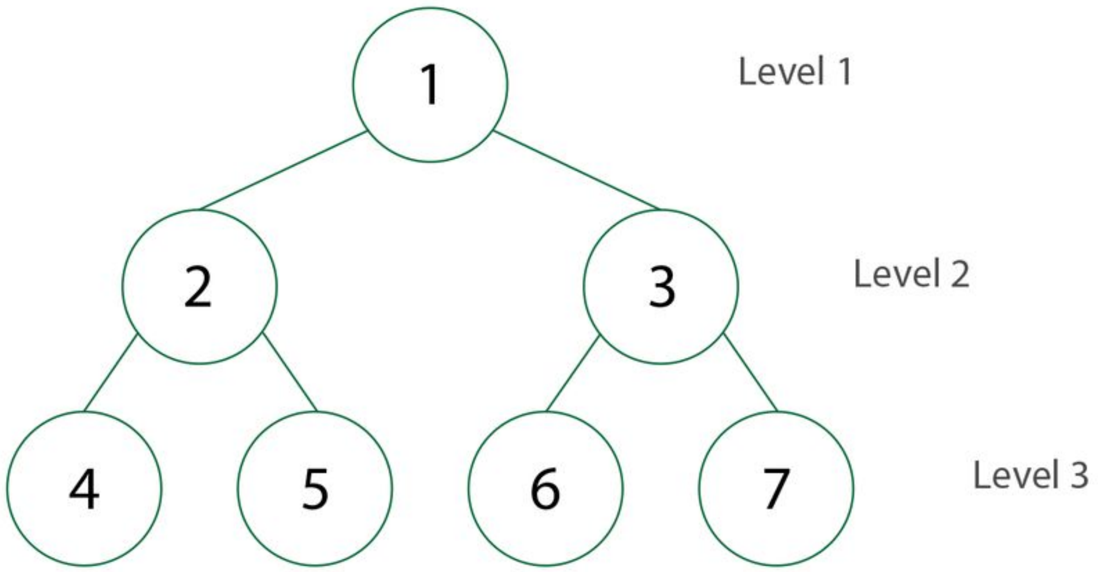
Uninformed Search (Blind)

- Main methods: Breadth First Search (BFS), Depth First Search (DFS).

- BFS = check houses row by row.
- DFS = search one street fully.

- Strength: Finds solution.
- Weakness: Wastes time & memory.

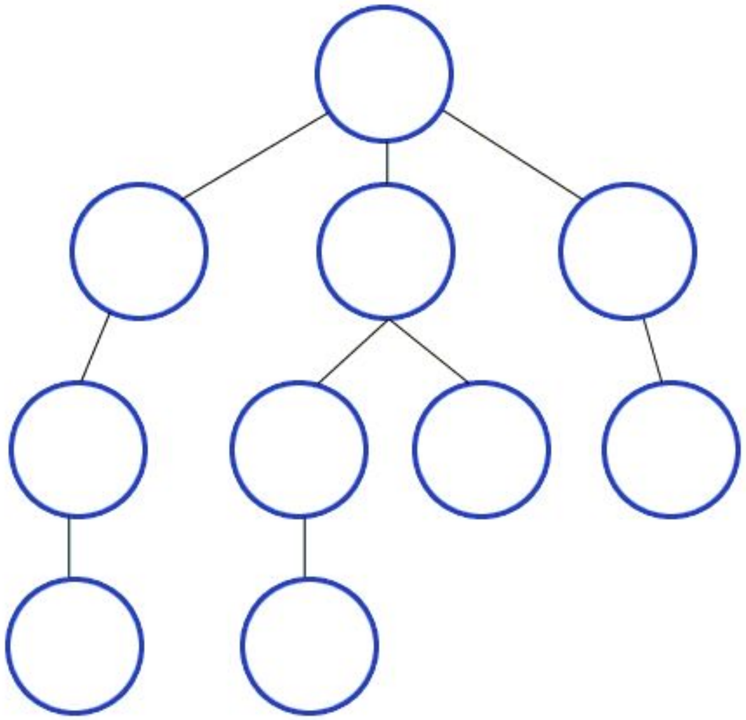
Types of Uninformed Search



- **Breadth-First Search (BFS)**
- Explores level by level.
- Always finds the **shortest path** if step cost = 1.
- **Example:**
  - You lost a key in a 2-story house. BFS = search ground floor room 1, then room 2, then move to floor 2.
- **Analogy:**
  Like water spreading in all directions equally.
- **Strengths:** Finds shortest path.
  Weakness: Uses huge memory if tree is wide.

Depth First Search (DFS)

- Goes deep into one branch before backtracking.
- Uses less memory.
- **Example:**
  - Searching a file in a computer → you open one folder, then sub-folder, then sub-sub-folder, until you reach dead end, then go back.
- **Analogy:**
  Like digging a well—you go straight down instead of wide.
- **Strengths:** Memory efficient.
  Weakness: May go infinitely deep and miss solution.



Uniform Cost Search (UCS)

Always expands the path with **lowest cost**.

Guarantees the **least expensive solution**.

Example: Traveling from Hyderabad to Karachi:

- Path 1: 3 hours but Rs. 2000 fuel.
- Path 2: 4 hours but Rs. 1000 fuel.
  UCS chooses Path 2 because it's cheaper.

Comparison of Uninformed Searches

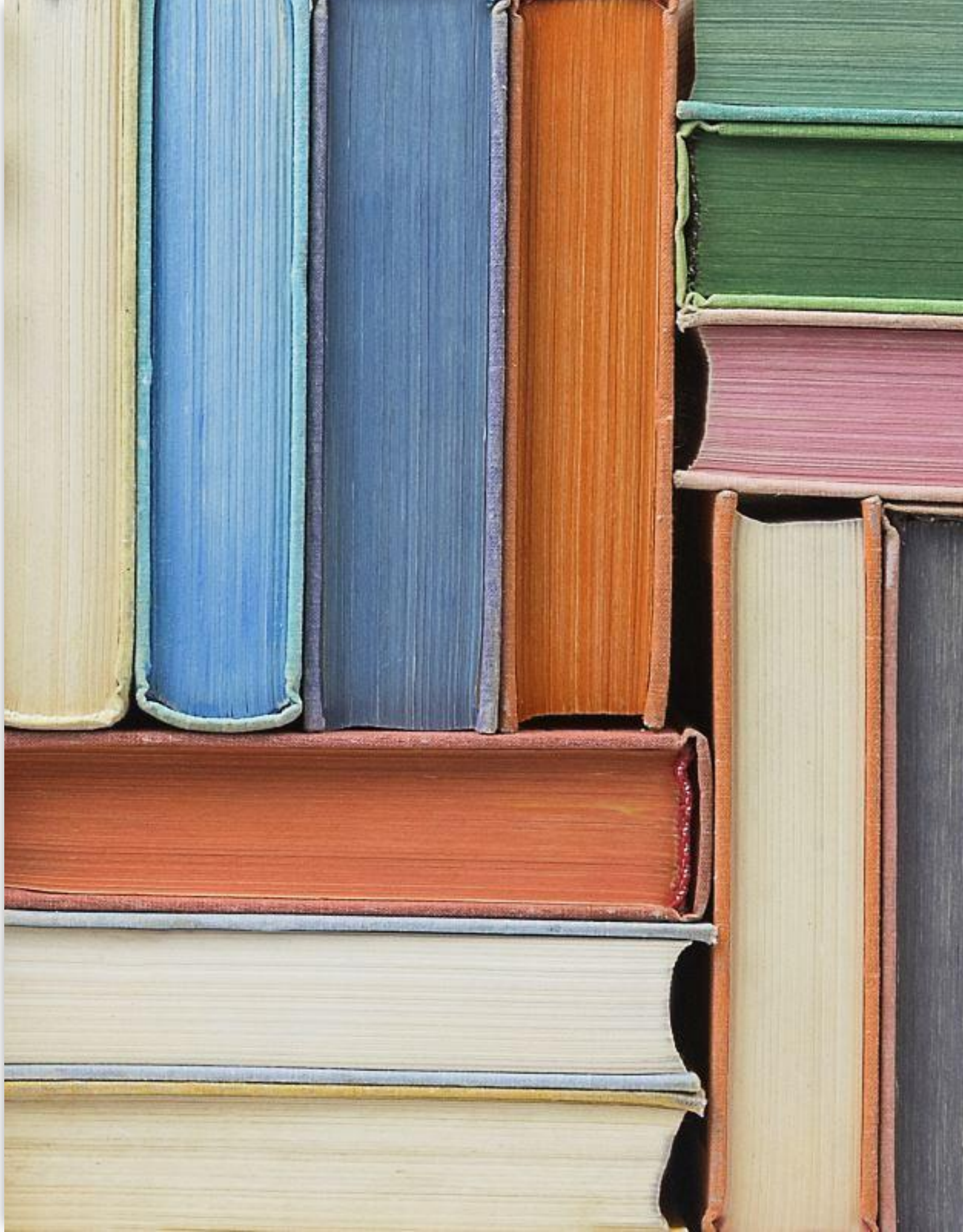
Algorithm	Strategy	Guarantee	Weakness
BFS	Expand level by level	Shortest path	High memory
DFS	Go deep then backtrack	Not guaranteed shortest	Can get stuck
UCS	Expand cheapest path	Optimal solution	Slow if costs vary

- Suppose you lost your **mobile phone at home**.
- **BFS:** Search each room one by one, completely covering each floor.
- **DFS:** Go to one room, check every corner, then next room.
- **UCS:** First check where you usually keep it (table, bed) before searching less likely spots.

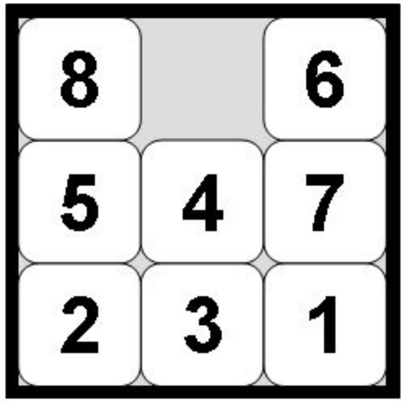
Informed Search

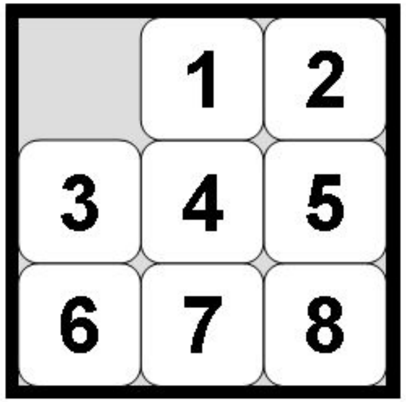
- Uses heuristics (h) to guide.
- Smarter than blind.

- Analogy: Blind = searching books one by one.
- Informed = using catalog.



Heuristics





- Heuristic = educated guess.
- Admissible heuristic = never overestimates.
- Example: 8-puzzle misplaced tiles.
- Map navigation = straight-line distance.
- Analogy: Cricket batsman estimating where ball lands.

Local Search

- Focus on goal, not path.
- Useful for optimization.

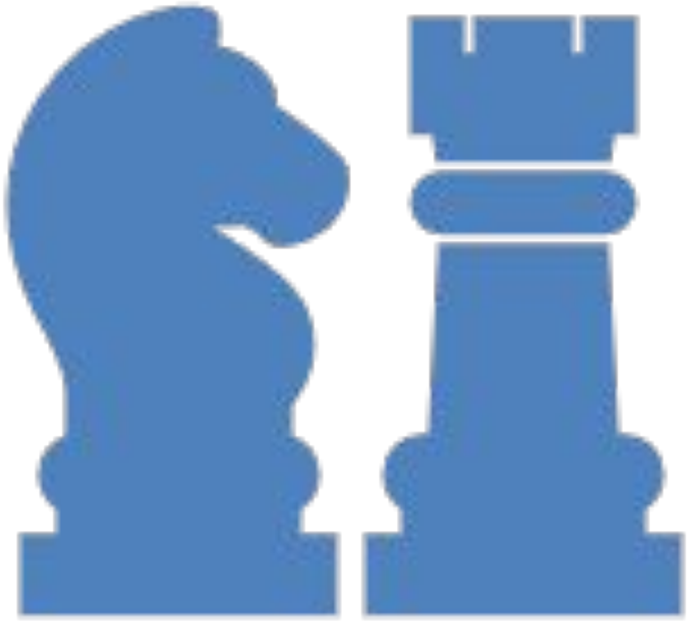
- Types: Hill Climbing, Simulated Annealing.

- Hill climbing (Climbing mountain in fog, only see nearby steps) = better offer step by step.

- Simulated annealing (Accepting low salary job now for future growth) = sometimes worse offer for better future.

Game Playing

- AI where the system must **compete against an opponent** in a strategic environment.
- Unlike normal problem-solving (which only has a goal), here the environment is **adversarial** – meaning there is another player trying to stop you.
- AI must **plan ahead** and assume the opponent is also smart.
- **Examples:** Chess, Ludo, Checkers, Carrom, and Tic-Tac-Toe.



Minimax Algorithm

- Algorithm used for decision-making in games.
- Idea: “Play as if your opponent is also playing perfectly.”
- MAX = maximize score.
- MIN = minimize score.
- Example: Tic-Tac-Toe → choose move that wins & blocks opponent.

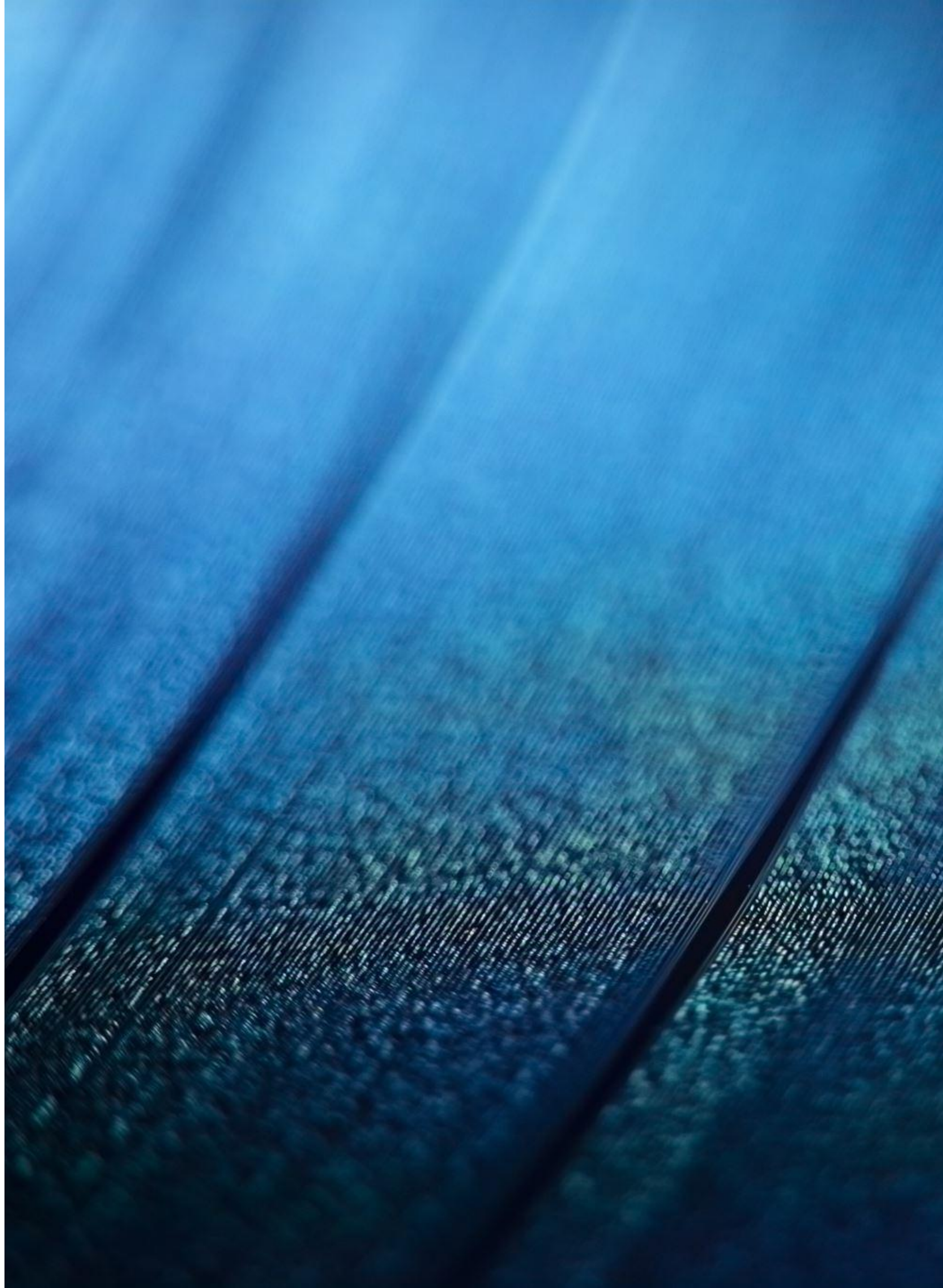
Alpha–Beta Pruning

Improves minimax.

Cuts useless branches.

Makes search much faster.

Analogy: If Student A has 90 marks, no need to check Student B with 70 when finding topper.



Summary

- Uninformed = Blind but complete.
- Informed = Uses heuristics.
- Heuristics = Educated guesses.
- Local Search = Goal only.
- Minimax = Best strategy in games.
- Alpha–Beta = Efficient minimax.