# Artificial Intelligence

## Week 5

**Dr Uzma Jamil**

**Department of Computer Science**

**Government College University, Faisalabad.**

# Search

- One of the most basic techniques in AI

- Can solve many problems that humans are not good at (achieving super-human performance)

- Very useful as a general algorithmic technique for solving many non-AI problems.

# How do we plan our holiday?

- **We must take into account various preferences and constraints to develop a schedule.**
- **An important technique in developing such a schedule is “hypothetical” reasoning.**

- Example: On holiday in Pakistan
- Currently in Faisalabad
- Flight leaves tomorrow from Lahore
  - Need plan to get to your plane
  - If I take a 6 am train where will I be at 2pm? Will I be still able to get to the airport on time?

# How do we plan our holiday?

- This kind of hypothetical reasoning involves asking

  **what state will I be in after taking certain actions, or after certain sequences of events?**

- From this we can reason about particular sequences of events or actions one should try to bring about to achieve a desirable state.

- Search is a computational method for capturing a particular version of this kind of reasoning.

# How do we plan our holiday?

- This kind of hypothetical reasoning involves asking

  **what state will I be in after taking certain actions, or after certain sequences of events?**

- From this we can reason about particular sequences of events or actions one should try to bring about to achieve a desirable state.

- Search is a computational method for capturing a particular version of this kind of reasoning.

# Many problems can be solved by search

# Why Search?

- Successful
  - Success in game playing programs based on search.
  - Many other AI problems can be successfully solved by search.
- Practical
  - Many problems don't have specific algorithms for solving them. Casting as search problems is often the easiest way of solving them.
  - Search can also be useful in approximation (e.g., local search in optimization problems).
  - Problem specific heuristics provides search with a way of exploiting extra knowledge.
- Some critical aspects of intelligent behaviour, e.g., planning, can be naturally cast as search.

# Limitations of Search

- There are many difficult questions that are not resolved by search. In particular, the whole question of how does an intelligent system formulate the problem it wants to solve as a search problem is not addressed by search.

- Search only shows how to solve the problem once we have it correctly formulated.

# Search

- Formulating a problem as search problem (representation)
- Heuristic Search
- Game-Tree-Search

- Readings
  - Introduction: Chapter 3.1 – 3.3
  - Uninformed Search: Chapter 3.4
  - Heuristic Search: Chapters 3.5, 3.6

# Representing a problem: The Formalism

To formulate a problem as a search problem we need the following components:

1. **STATE SPACE:** Formulate a state space over which we perform search. The state space is a way or representing in a computer the states of the real problem.
2. **ACTIONS or STATE SPACE Transitions:** Formulate actions that allow one to move between different states. The actions reflect the actions one can take in the real problem but operate on the state space instead.

# Representing a problem: The Formalism

To formulate a problem as a search problem we need the following components:

3. **INITIAL or START STATE and GOAL:** Identify the initial state that best represents the starting conditions, and the goal or condition one wants to achieve.
4. **Heuristics:** Formulate various heuristics to help guide the search process.

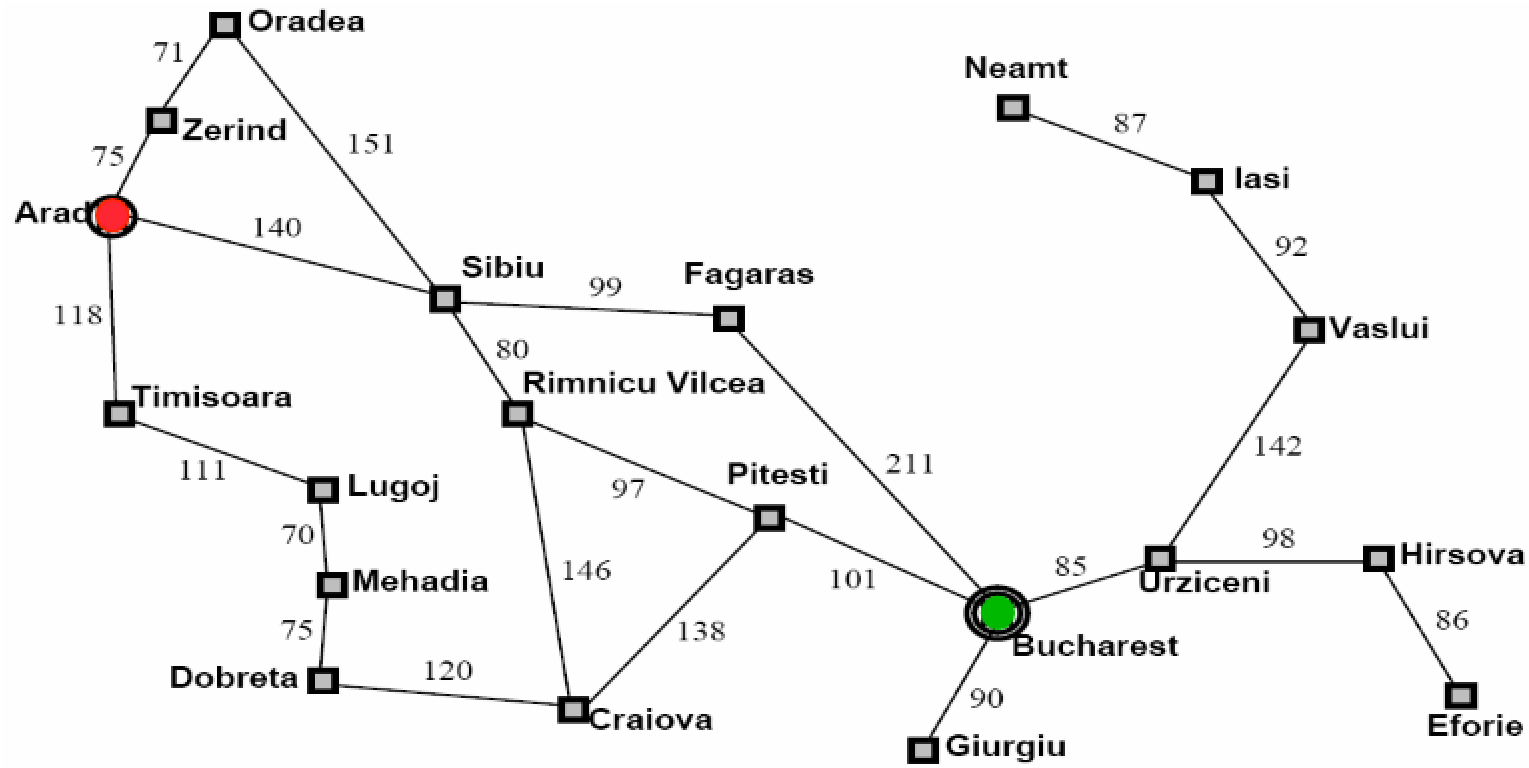
# The Formalism

Once the problem has been formulated as a state space search, various algorithms can be utilized to solve the problem.

- A solution to the problem will be a sequence of actions/moves that can transform your current state into a state where your desired condition holds.

# Example 1: Romania Travel.

Currently in Arad, need to get to Bucharest by tomorrow to catch a flight. What is the State Space?



# Example 1.

- State space.
  - States: the various cities you could be located in.
    - Our abstraction: we are ignoring the low level details of driving, states where you are on the road between cities, etc.
  - Actions: drive between neighboring cities.
  - Initial state: in Arad
  - Desired condition (Goal): be in a state where you are in Bucharest. (How many states satisfy this condition?)
- Solution will be the route, the sequence of cities to travel through to get to Bucharest.

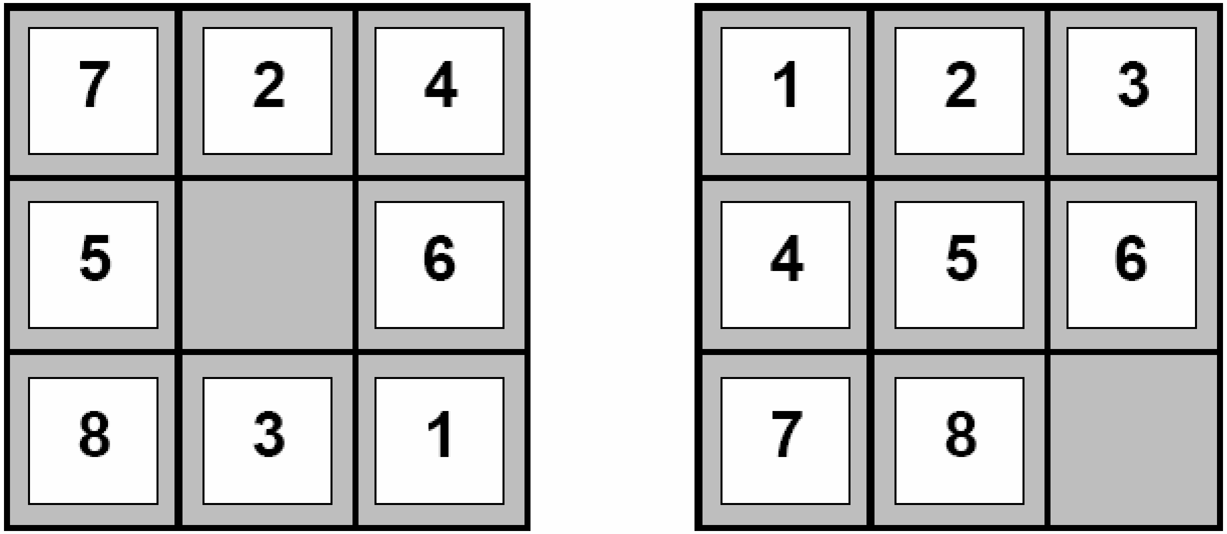
# Example 2.

- Water Jugs
  - We have a 3 gallon (liter) jug and a 4 gallon jug. We can fill either jug to the top from a tap, we can empty either jug, or we can pour one jug into the other (at least until the other jug is full).
  - States: pairs of numbers (gal3, gal4) where gal3 is the number of gallons in the 3 gallon jug, and gal4 is the number of gallons in the 4 gallon jug.
  - Actions: Empty-3-Gallon, Empty-4-Gallon, Fill-3-Gallon, Fill-4-Gallon, Pour-3-into-4, Pour 4-into-3.
  - Initial state: Various, e.g., (0,0)
  - Desired condition (Goal): Various, e.g., (0,2) or (*, 3) where * means we don't care.

# Example 2.

- Water Jugs
  - If we start off with gal3 and gal4 as integer, can only reach integer values.
  - Some values, e.g., (1,2) are not reachable from some initial state, e.g., (0,0).
  - Some actions are no-ops. They do not change the state, e.g.,
    - (0,0) → Empty-3-Gallon → (0,0)

# Example 3. The 8-Puzzle




Start State    Goal State

Rule: Can slide a tile into the blank spot.
Alternative view: move the blank spot around.

# Example 3. The 8-Puzzle

- State space.
  - States: The different configurations of the tiles. How many different states?
  - Actions: Moving the blank up, down, left, right. Can every action be performed in every state?
  - Initial state: e.g., state shown on previous slide.
  - Desired condition (Goal): be in a state where the tiles are all in the positions shown on the previous slide.
- Solution will be a sequence of moves of the blank that transform the initial state to a goal state.

# Example 3. The 8-Puzzle

- Although there are 9! different configurations of the tiles (362,880) in fact the state space is divided into two disjoint parts.
- Only when the blank is in the middle are all four actions possible.
- Our goal condition is satisfied by only a single state. But one could easily have a goal condition like
  - The 8 is in the upper left hand corner.
    - How many different states satisfy this goal?

# More complex situations

- Perhaps actions lead to multiple states, e.g., flip a coin to obtain heads OR tails. Or we don't know for sure what the initial state is (prize is behind door 1, 2, or 3). Now we might want to consider how likely different states and action outcomes are.
- This leads to probabilistic models of the search space and different algorithms for solving the problem.
- Later we will see some techniques for reasoning under uncertainty.