اللَّهُمَّ لا سَهْلَ إلاَّ ما جَعَلْتَهُ سَهْلاً
وأنْتَ تَجْعَلُ الحَزْنَ إذا شِئْتَ سَهْلاً

IDSS

PART (4)

Search Agent

بسم الله الرحمن الرحيم

شبابنا الحلو .. ربنا معاكوا في اخر ترم وفي مشروع التخرج.

حنكون معاكوا ان شاء الله في ٣ مواد ... متشغلوش بالكوا بيهم و ركزوا في مشروعكوا

صلي علي سيدنا محمد

Agents

1. Reflex Agent

Use a mapping **from states** to **actions**. [If-then]

Considers how the world *is*:

- Choose action based on **current percept.**
- **Do not consider the future** consequences of actions.
- **Example**: A thermostat that turns on the heater when the temperature drops below a certain threshold is a reflex agent.

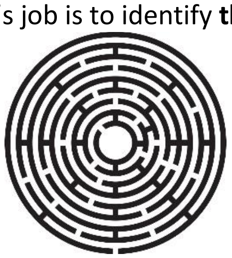
2. Planning (Goal-based) agent

problem solving agents or **planning agents.**

- Considers how the world *WOULD BE*:

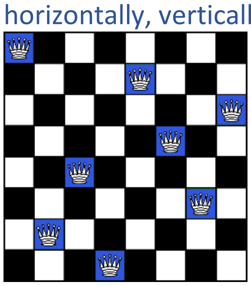
- Decisions based on (hypothesized) **consequences of actions.**
- Must have a model of how the world evolves in response to actions.
- Must formulate a goal.

- Agents that **work towards a goal**.
- Agents consider **the impact of actions on future states**.
- Agent's job is to identify **the action or series of actions that lead to the goal**.



- Agents that **work towards a goal**.

The 8-queen problem: on a chess board, place 8 queens so that no queen is attacking any other horizontally, vertically or diagonally.



Number of possible sequences to investigate:
$64 * 63 * 62 * ... * 57 = 1.8 \times 10^{14}$

- Agents consider the impact of actions on future states.
- Agent's job is to identify the action or series of actions that lead to the goal.
- Formalized as a search through possible solutions.

صلي علي سيدنا محمد

Problem solving as search:

(a) Goal formulation

(b) Problem formulation

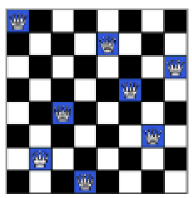
Problem formulation:

- **Initial state:** the state in which the agent starts
- **States:** All states reachable from the initial state by any sequence of actions (**State space**)
- **Actions:** possible actions available to the agent. At a state s, $Actions(s)$ returns the set of actions that can be executed in state s. (**Action space**)
- **Transition model:** A description of what each action does. $Results(s, a)$
- **Goal test:** determines if a given state is a goal state.
- **Path cost:** function that assigns a numeric cost to a path w.r.t. performance measure.

Example:

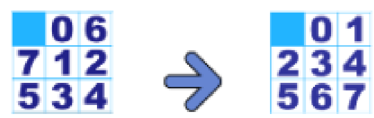
consider the problem of designing goal-based agents in **fully observable**, **deterministic**, **discrete**, **known** environments.

1



- **States:** all arrangements of 0 to 8 queens on the board.
- **Initial state:** No queen on the board
- **Actions:** Add a queen to any empty square
- **Transition model:** updated board
- **Goal test:** 8 queens on the board with none attacked

2



- **States:** Location of each of the 8 tiles in the 3x3 grid
- **Initial state:** Any state
- **Actions:** Move Left, Right, Up or Down
- **Transition model:** Given a state and an action, returns re- sulting state
- **Goal test:** state matches the goal state?
- **Path cost:** total moves, each move costs 1.

3

-On vacation in Romania; currently in Arad.

-Flight leaves tomorrow from Bucharest.

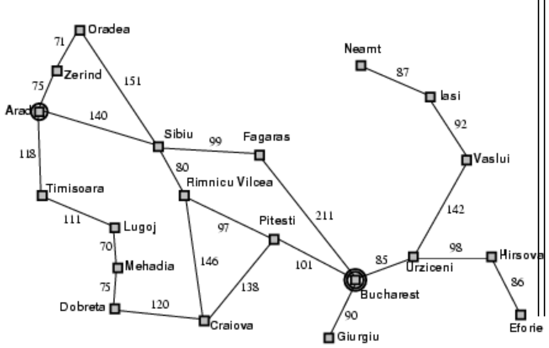
Initial state
- Arad

Actions
- Go from one city to another

Transition Model
- If you go from city A to city B, you end up in city B

Goal State
- Bucharest

Path Cost
- Sum of edge costs *(total distance)*



4

Initial state

Actions

Steps

Transition model

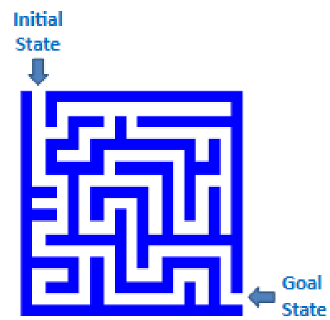
What state results from performing a given action in each state?

Goal state

Solution Path

Path cost

Assume that it is a sum of nonnegative *step costs*

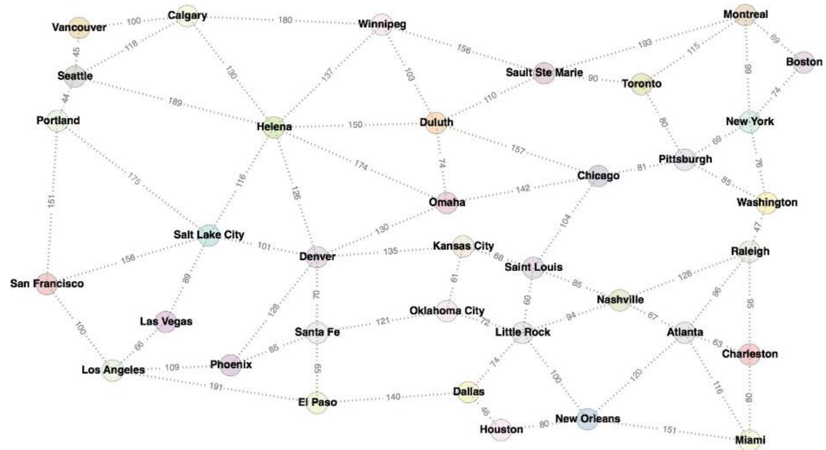


The **optimal solution** is the sequence of actions that gives the *lowest* path cost for reaching the goal.

صلي علي سيدنا محمد

5

- **States:** In City where
  City ∈ {Los Angeles, San Francisco, Denver,...}
- **Initial state:** In Boston
- **Actions:** Go New York, etc.
- **Transition model:**
  Results (In (Boston), Go (New York)) = In(New York)
- **Goal test:** In(Denver)
- **Path cost:** path length in kilometers



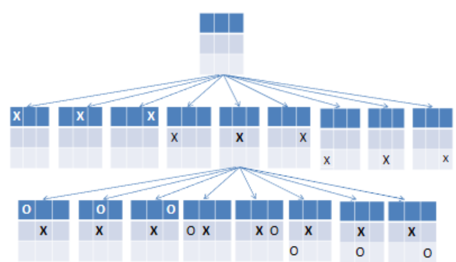
Real Example:

Route finding problem: typically, our example of map search, where we need to go from location to location using links or transitions. Examples of applications include tools for driving directions in websites, in-car systems, etc.

State Space Search

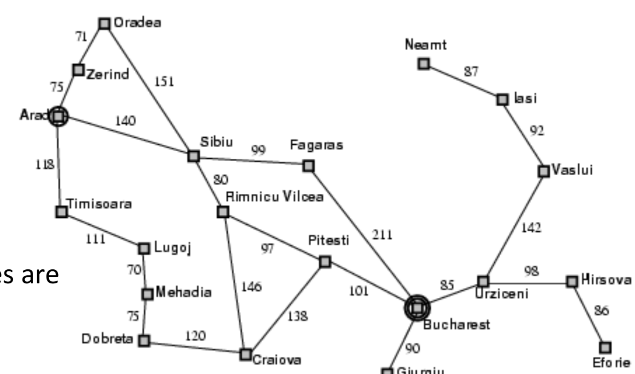
Problems are solved by searching among alternative choices (Start → Goal).

- Humans consider **several alternative strategies** on their **way to solving a problem.**
  - A Chess player considers a few alternative moves.
  - A mathematician chooses from a different strategy to find proof for a theorem.
  - A physician evaluates several possible diagnoses.
  - **Tic-Tac-Toe Game**



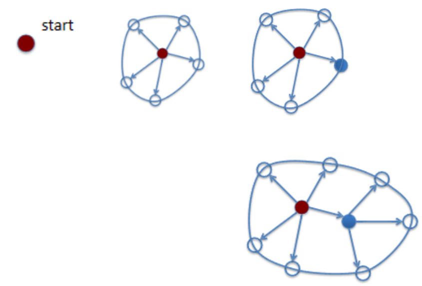
- Human beings **do not search the entire state space** (exhaustive search).
- Only alternatives **that experience has shown to be effective** are explored.
- Human problem solving is based **on judgmental rules** that **limit the exploration of search space** to those portions of state space that seem somehow promising.
- These judgmental rules are known as **"heuristics"**.

- **Example: -**
  The initial state, actions, and transition model define the state space of the problem;
- **The set of all states reachable from the initial state** by any sequence of actions.
- Can be represented as a directed graph where the nodes are states and links between nodes are actions.



- **An AI problem can be represented as a *state space graph.***
- **A graph is a set of *nodes* and *links* that connect them.**

Search: Basic idea



- **Let's begin at the start state and expand it by making a list**
- **of all possible successor states.**
- **Maintain a frontier or a list of unexpanded states.**
- **At each step, pick a state from the frontier to expand.**
- **Keep going until you reach a goal state.**
- **Try to expand as few states as possible.**

State space vs. Search space.

صلي علي سيدنا محمد

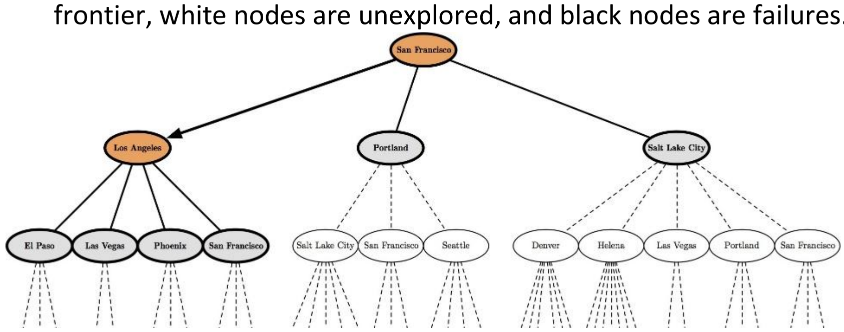
- **State space**: a *physical* configuration
- **Search space**: an *abstract* configuration
  search tree or graph of possible solutions.
  - **Search tree: models the sequence of actions**
    - Root: initial state
    - Branches: actions
    - Nodes: results from actions. A node has: parent, children, depth, path cost, associated state in the state space.
  - **Expand:** A function that given a node, creates all children nodes

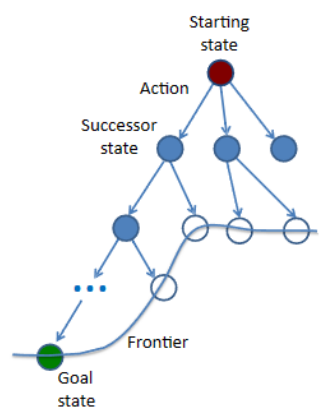
Example (Search Tree): -

The search space is divided into three regions:

1. **Explored (a.k.a. Closed List, Visited Set)**
2. **Frontier (a.k.a. Open List, the Fringe)**
3. **Unexplored.**

- The essence of search is moving nodes from regions **(3) to (2) to (1),** and the essence of search strategy is deciding the order of such moves.
- In the following we adopt the following color coding: orange nodes are explored, grey nodes are the frontier, white nodes are unexplored, and black nodes are failures.





- **<u>Tree Search Algorithm Outline</u>**
- **The root node** corresponds to the **starting state.**
- **The children of a node** correspond to **the successor states** of that node's state.
- **A path** through the tree corresponds to a sequence of actions.
- **A solution** is a path ending in the goal state.
- A state is a representation of the world, while a node is a data structure that is part of the search tree

- **Initialize the frontier using the starting state**. While the frontier is not empty:
  1. **Choose a frontier node** according to **search strategy** and take it off the frontier.
  2. If the node contains the **goal state**, **return solution.**
  3. **Else expand the node** and add its children to the frontier.