CENG 466
Artificial Intelligence

Lecture 3

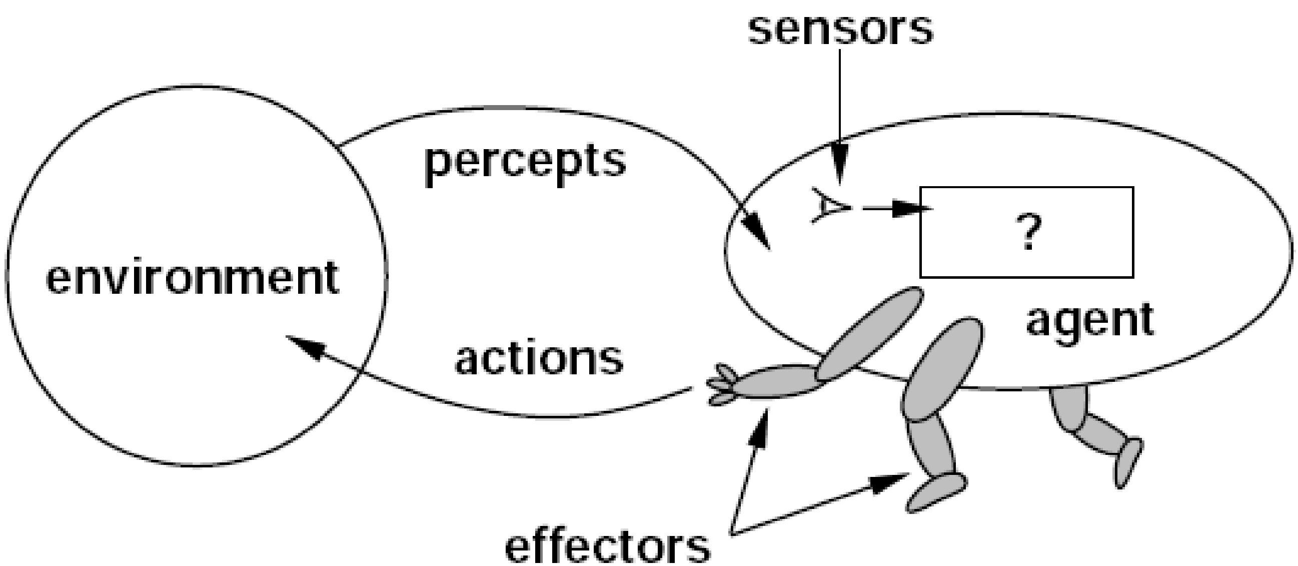
Solving Problems by Searching (I)

Topics

- Example Intelligent Agents
- Goal Formulation
- Problem Formulation
- States and State-Spaces
- Well-defined Problems
- Searching

Intelligent Agents

- An agent is something that perceives and acts in an environment
- An ideal agent always takes actions that maximizes its performance
- An agent adopts a goal and searches the best path to reach that goal



Rational Agent

- A rational agent is an agent which does the right action
- The right action will cause the agent to be most successful

Agent Types

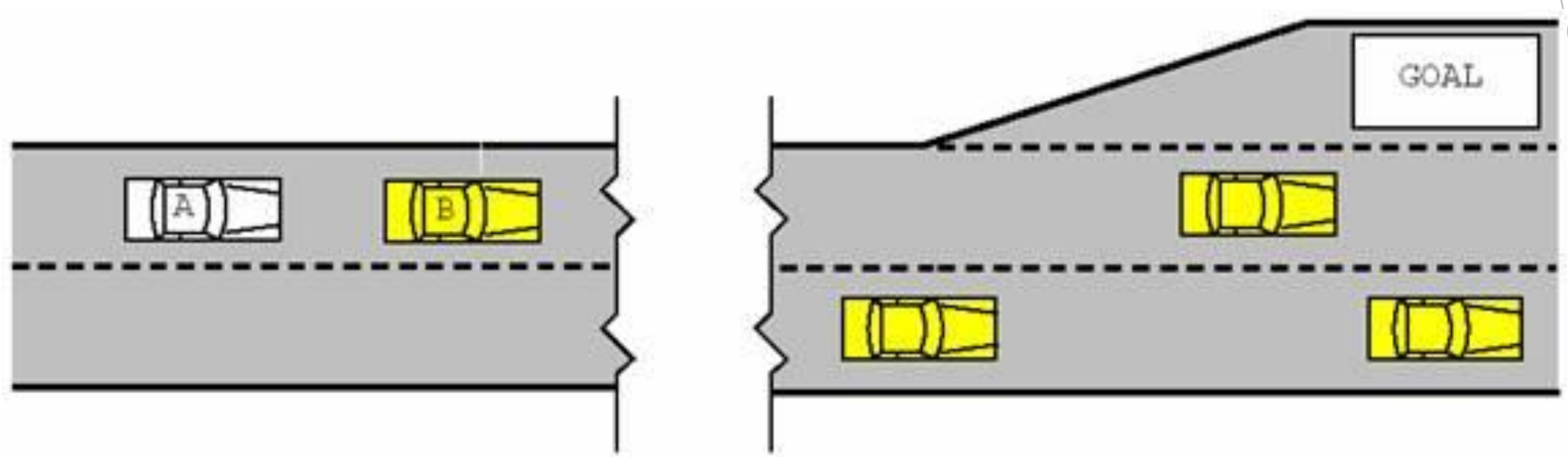
- Simple Reflex agent
- Agents that can remember
- Goal-based agents
- Utility-based agents

Some Examples of Intelligent Agents (1)

Automated driving

- This is a utility based agent.
- **Percepts**: Cameras are used to gain positions of car, the edges of lanes, and the position of the goals.
- **Actions**: The car can speed up, slow down, change lanes, turn, park, pull away

Some Examples of Intelligent Agents (1)



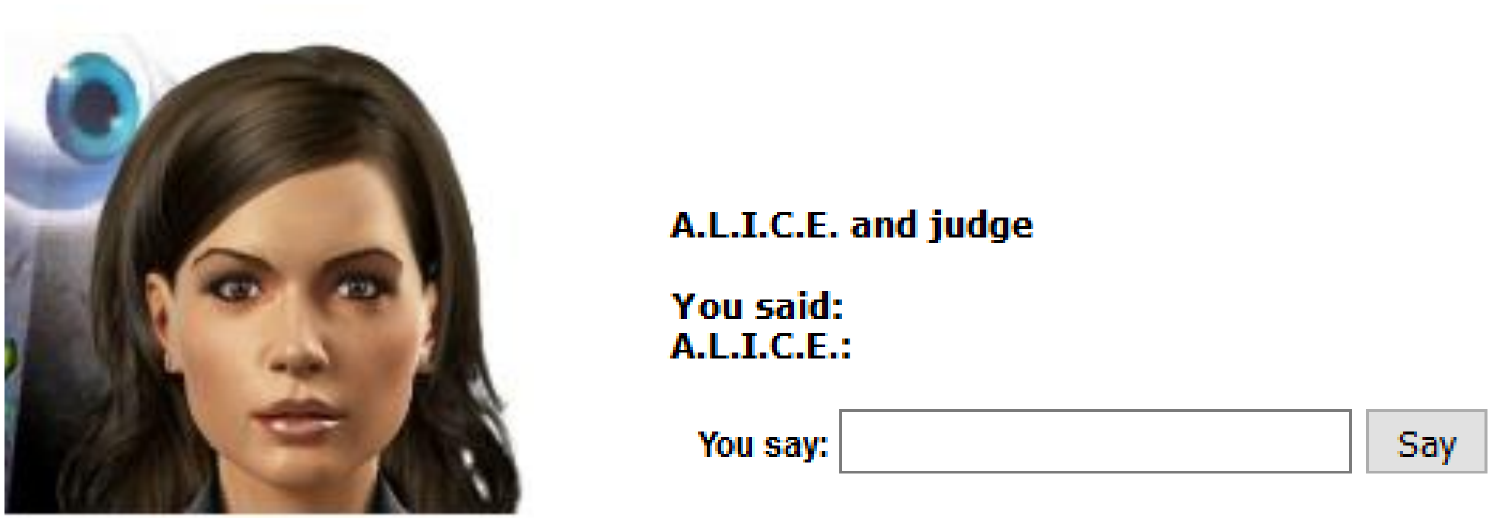
http://www.ri.cmu.edu/research_project_view.html?menu_id=261

Some Examples of Intelligent Agents (2)

A.L.I.C.E. – an online chat bot

- A.L.I.C.E. is a learning reflex agent.
- Given certain keywords the bot has a pre-determined response
- But if the bot doesn't know how to reply then it will ask a question which will help it to learn the answer to the question.
- This builds up a large amount of condition-action rules.

Some Examples of Intelligent Agents (2)



- http://www.pandorabots.com/pandora/talk?botid=f5d922d97e345aa1

Some Examples of Intelligent Agents (3)

Goal-Oriented Web Search User Interfaces

- GOOSE (GOal-Oriented Search Engine interface) is an adaptive search engine interface that uses natural language processing
- It parses a user’s search goal, and uses "common sense" reasoning to interpret this goal.
- Then it uses the goal to create an effective query.

Some Examples of Intelligent Agents (3)

http://agents.media.mit.edu/projects.html

Problem Solving Agents

- Intelligent agents are supposed to act in such a way that their performance measure is maximized.

- Problem-solving agents decide what to do by finding sequences of actions that lead to their desirable states (goals).

- Therefore, the agent
  - defines the goals and
  - finds the path to reach that goal

Goal Formulation

- An intelligent agent should define its goals together with the limiting factors. This is called **Goal Formulation**
- Example:
- If the goal is going to a specific city, the limiting factors can be:
  - Time to arrive,
  - The cities/roads that should be avoided
  - The maximum distance of the travel
- Goal Formulation is the first step in problem solving.

Problem Formulation

- To reach the goal, the agent should decide about the necessary actions
- **Problem formulation** is the process of deciding what actions and states to consider to reach the goal
- Problem Formulation follows goal formulation.

States and State-Spaces

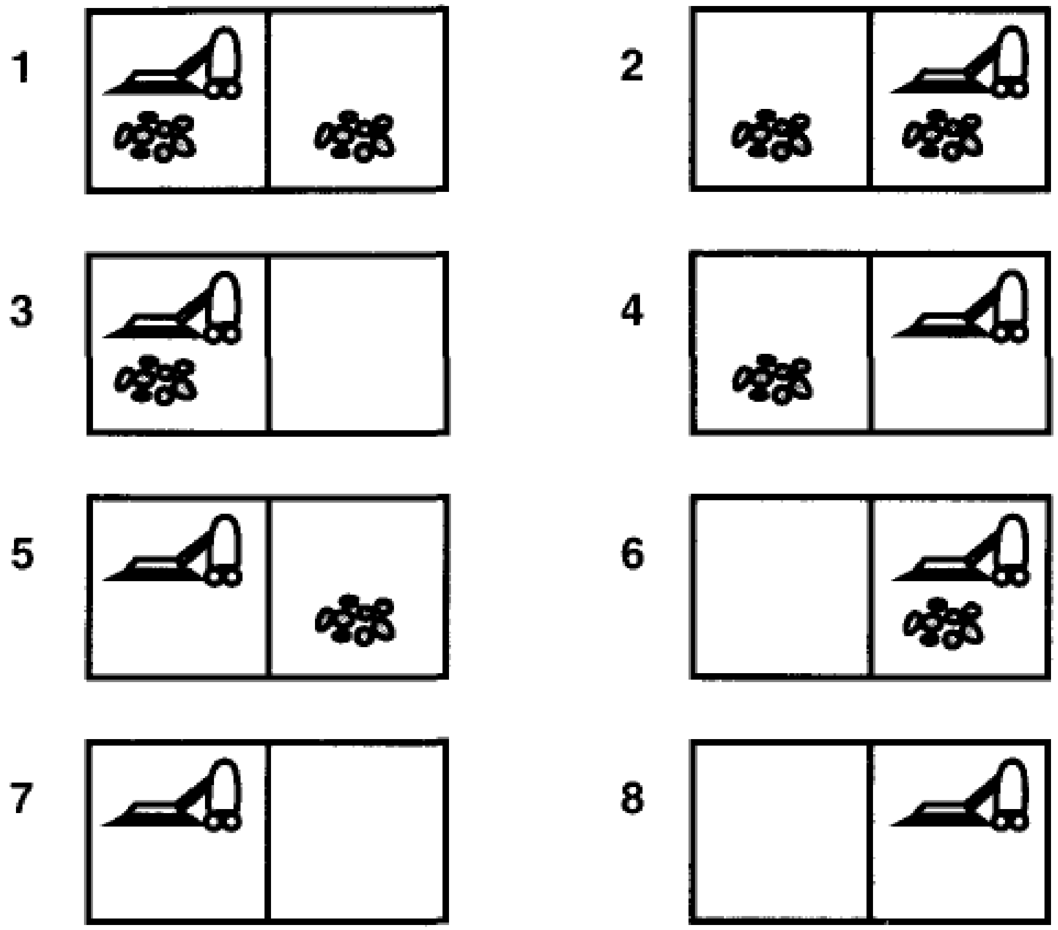
- State: The set of all information items that describe a system at a given time.
- State Space: a set of states that a problem can be in.
- The set of states forms a graph where two states are connected if there is an action that can be performed to transform the first state into the second.

States and State-Spaces

- State space is the set of states that an intelligent agent can be in.
- An action takes the agent from one state to another one.
- State space search is finding a sequence of states starting from the initial state to the goal
- Searching in AI differs from traditional computer science search methods because the state space is often implicit (too big to fit into the memory)

Example 1

Assume a vacuum cleaner can be in one of the two possible positions. The state space is:



Example 1 (cont.)

- The actions available are:
  - Left : The agent goes to the left side room
  - Right : The agent goes to the right side room
  - Clean : The agent cleans the current place

Example 1 (cont.)

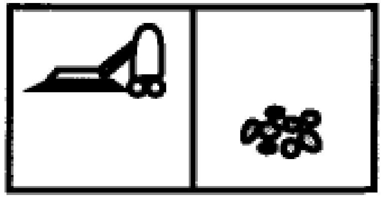
- Case 1: The vacuum cleaner cannot percept its environment.
- Problem formulation:

  (Left, Clean, Right, Clean)

Example 1 (cont.)

- Case 2: The vacuum cleaner can percept the environment.
- The cleaner formulates the problem based on its current state
- E.g. Current state is

5



then action is [Right, Clean]

Well-defined Problems

- A problem is well-defined if we can determine the followings:

1. **The initial state**
2. The set of **possible actions** available to the agent.
3. **The goal test:** The agent can apply to a single state to determine if it is a goal state.
4. **A path cost** function: A function that assigns a cost to a path.

Measuring Problem Solving Performance

- The effectiveness of a search can be measured in at least three ways.

1. First, does it find a solution at all?
2. Second, is it a good solution (one with a low path cost)?
3. Third, what is the **search cost?** (the time and memory required to find a solution)

- The **total cost** of the search is the sum of the path cost and the search cost.

Choosing States and Actions

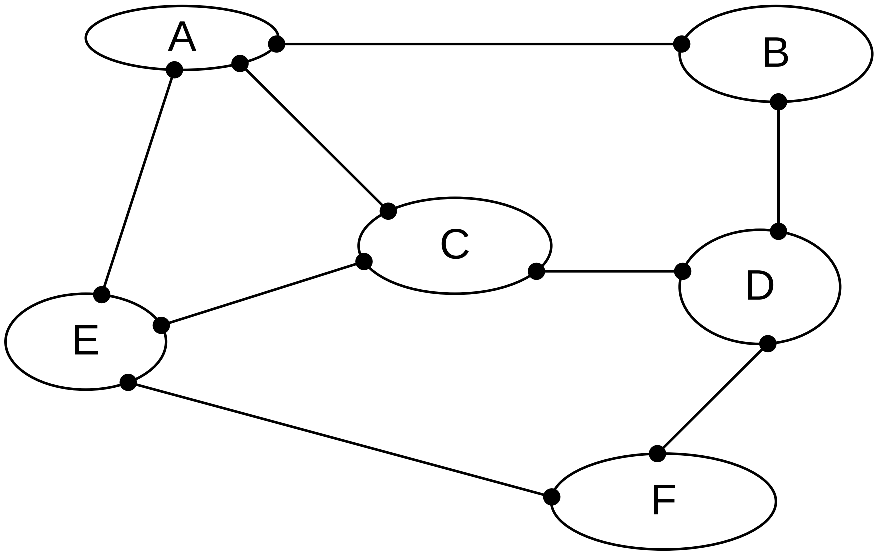
- An intelligent agent can decide what to do by first examining the different possible *sequences* of actions, and then choosing the best one.
- This process of looking for such a sequence is called **search**.
- A search algorithm takes a problem as input and returns a **solution** in the form of an action sequence.

Executing Solutions

- When a solution is found, the actions it recommends can be carried out. This is called the execution phase.
- Thus, we have a simple "formulate, search, execute" design for the agent

Example 2

- **Travelling Salesperson Problem:** Assume N cities are given in a graph. Visit every city in this graph at least once, starting and ending in the first city.
- E.g. Cities { A, B, C, D, E, F}



- **States: Current city and the list of cities visited so far**
  - **Initial state = A**
  - **Other states:, AB, AC, ABD, ACE, ABDF,….**

Example 2 (Cont.)

- The traveling Salesperson problem can be modified by adding distances between cities.
- A cost function can compute the path cost and total cost using these distance values

Searching

- Assuming that the agent knows:
  - how to define a problem,
  - how to recognize a solution (goal),
- finding a solution is done by a search through the state space.

Data Structures

- In the search process we will build up a **search tree**
  - **Root of the tree is the initial state**
  - Each node of this tree is a state in the state space
  - The leaf nodes of the tree correspond to states that do not have successors in the tree, because:
    - They have not been expanded yet
    - They were expanded, but generated the empty set.

Search Categories

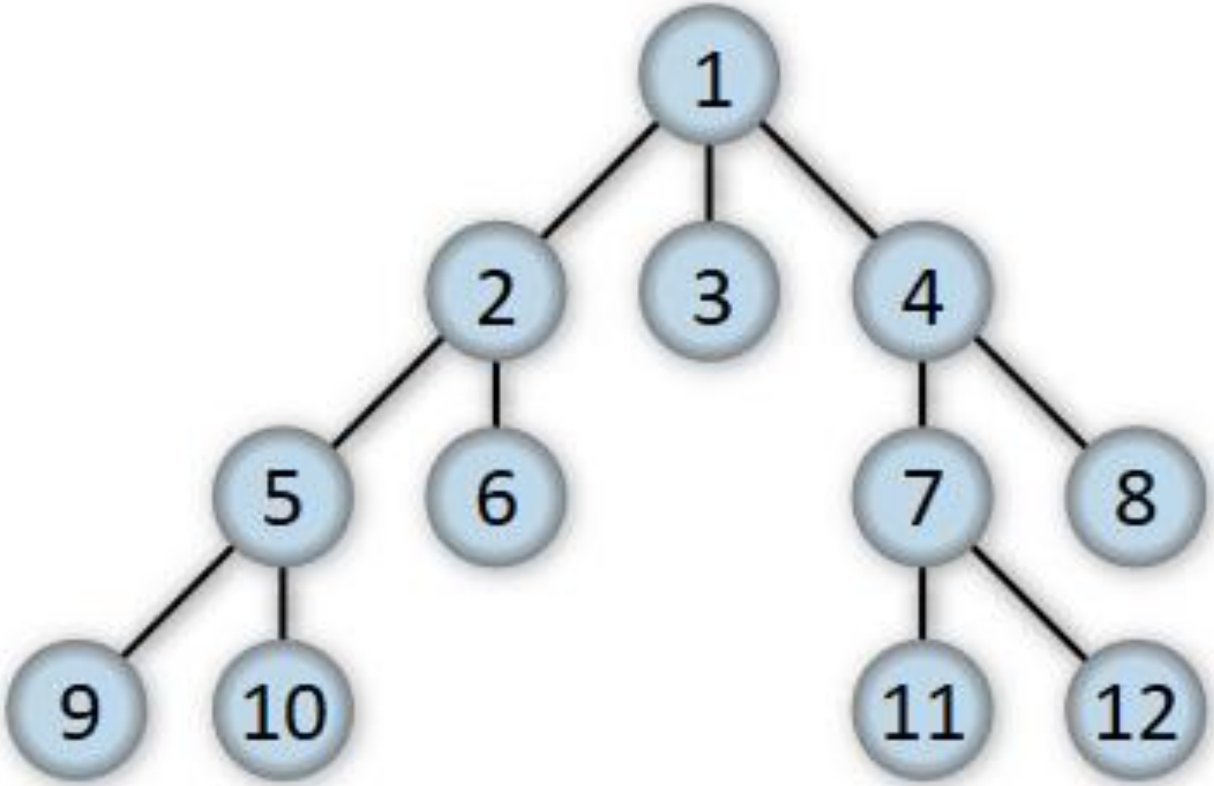
- Un-informed Searches: If we have no extra information about the problem

- Informed Searches: If we have extra information about the problem.

Breadth-First Search

- In breadth first search we start from the root node
- Then visit all nodes at distance 1 from the root
- Next all nodes at distance 2 from the root
- Until
  - A goal is found
  - All nodes are visited

Example



When to Use Breadth First Search?

- When the search tree (state space) is too big.
  - e.g. Playing Chess
- When a close solution is expected

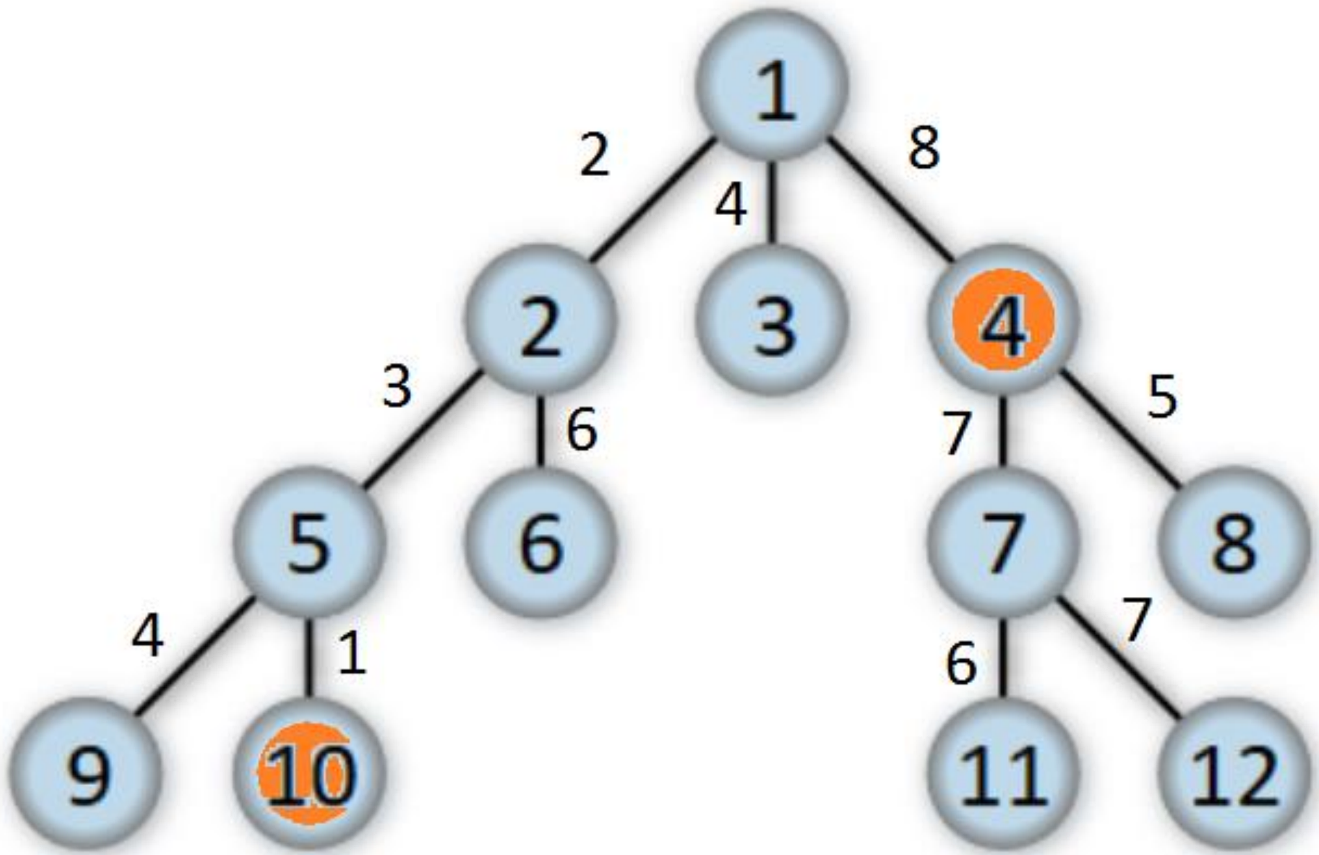
Uniform Cost Search

- Breadth-first search finds the *shallowest* goal state, but this may not always be the least-cost solution.
- **Uniform cost search** modifies the breadth-first strategy by always expanding the lowest-cost node.

Uniform Cost Search Examples

- Assume a search tree has multiple goal states.
- Each link is labeled with a link-cost value
- Initial node is the root node.

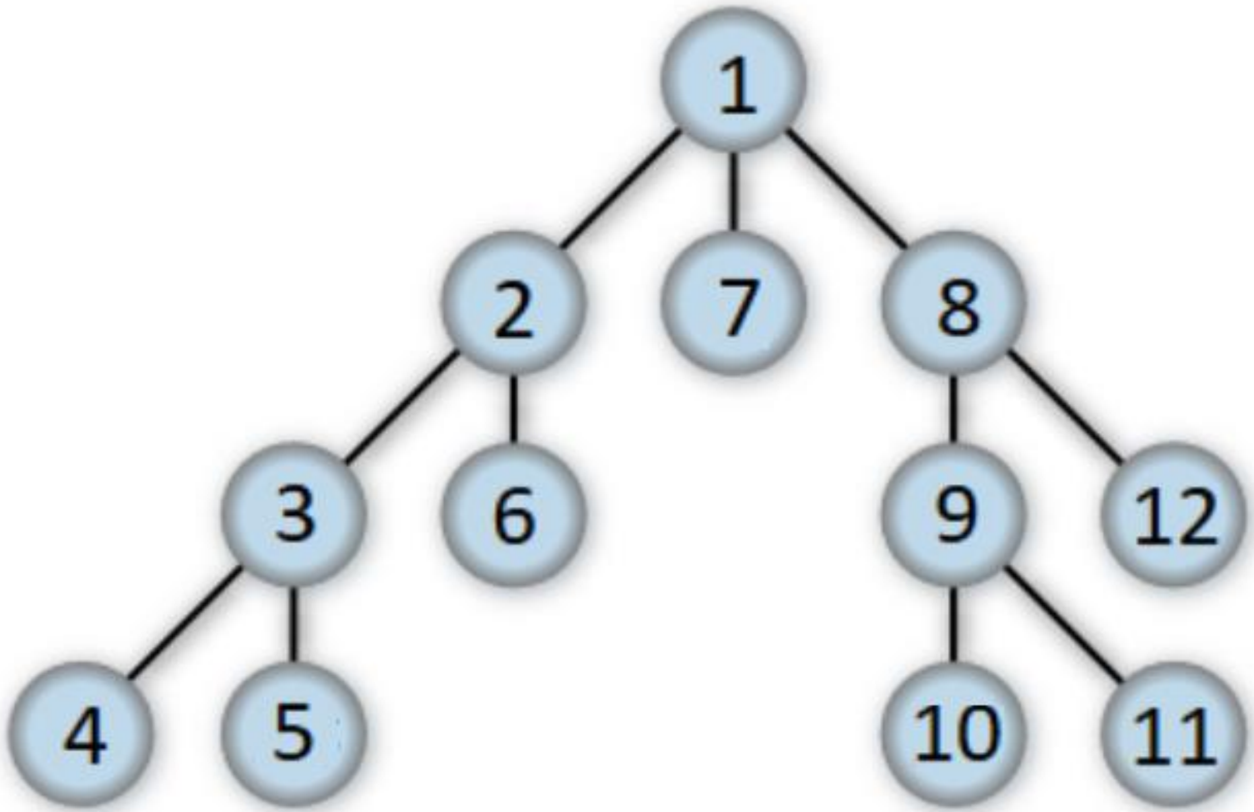
Uniform Cost Search Examples



Depth First Search

- Depth-first search always expands one of the nodes at the deepest level of the tree.
- When the search reaches a non-goal node with no child node, the search goes back and expands nodes at upper levels.

Depth First Search Example



Questions?