CENG 466
Artificial Intelligence

Lecture 4

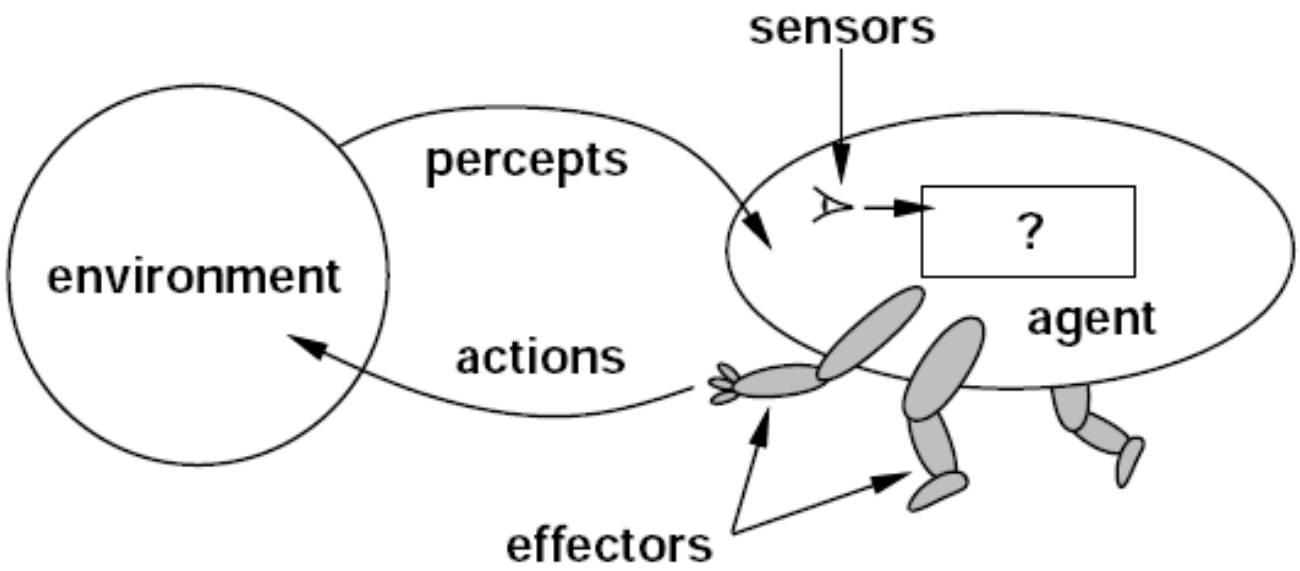
Solving Problems by Searching (II)

Topics

- Search Categories
- Breadth First Search
- Uniform Cost Search
- Depth First Search
- Depth Limited Search
- Iterative Deepening Search
- Bidirectional Search
- Constraint Satisfaction Search

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



States and State-Spaces

- State: The set of all information items that describe a system at a given time.
- 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

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.

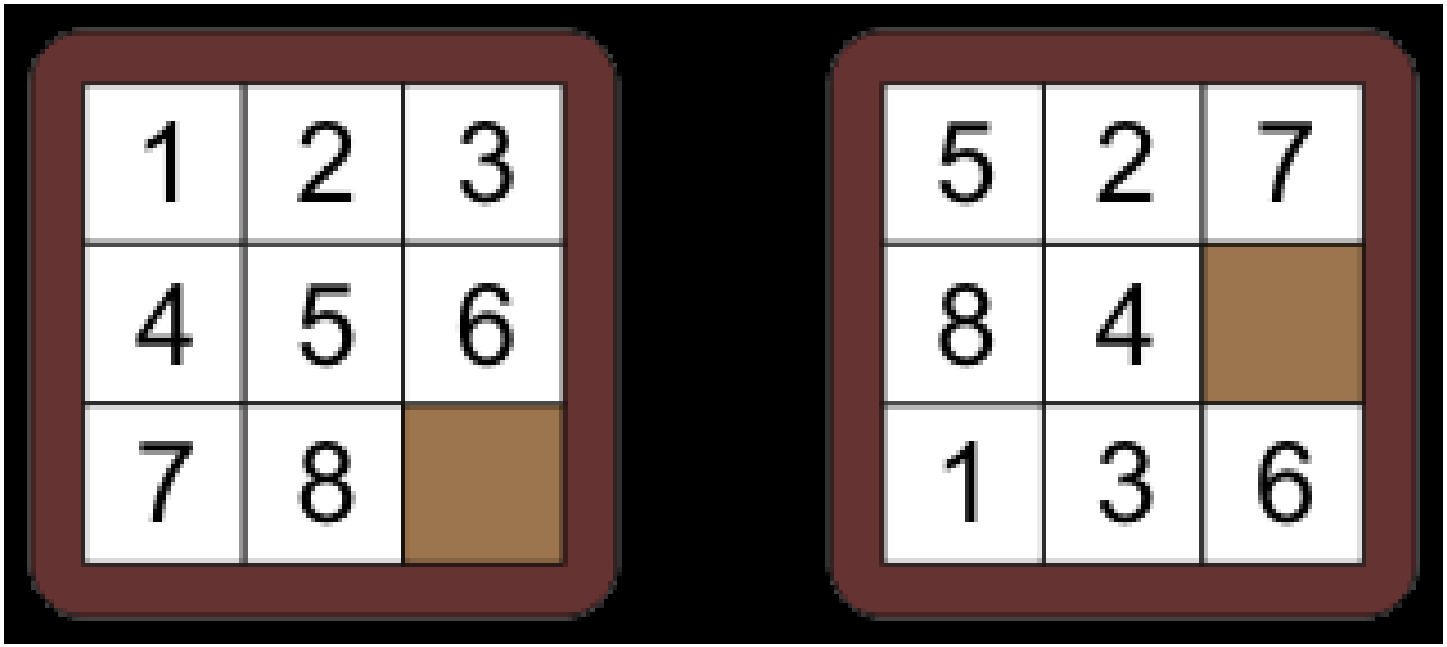
Searching

- 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.

8-Puzzle Example

- A puzzle with 8 numbers (1 through 8) on 8 tiles and 1 empty place is given
- The goal is putting the numbers in order.
- Initially the numbers are distributed randomly.
- Actions: A tile can be slid into an adjacent empty place

8-Puzzle Example



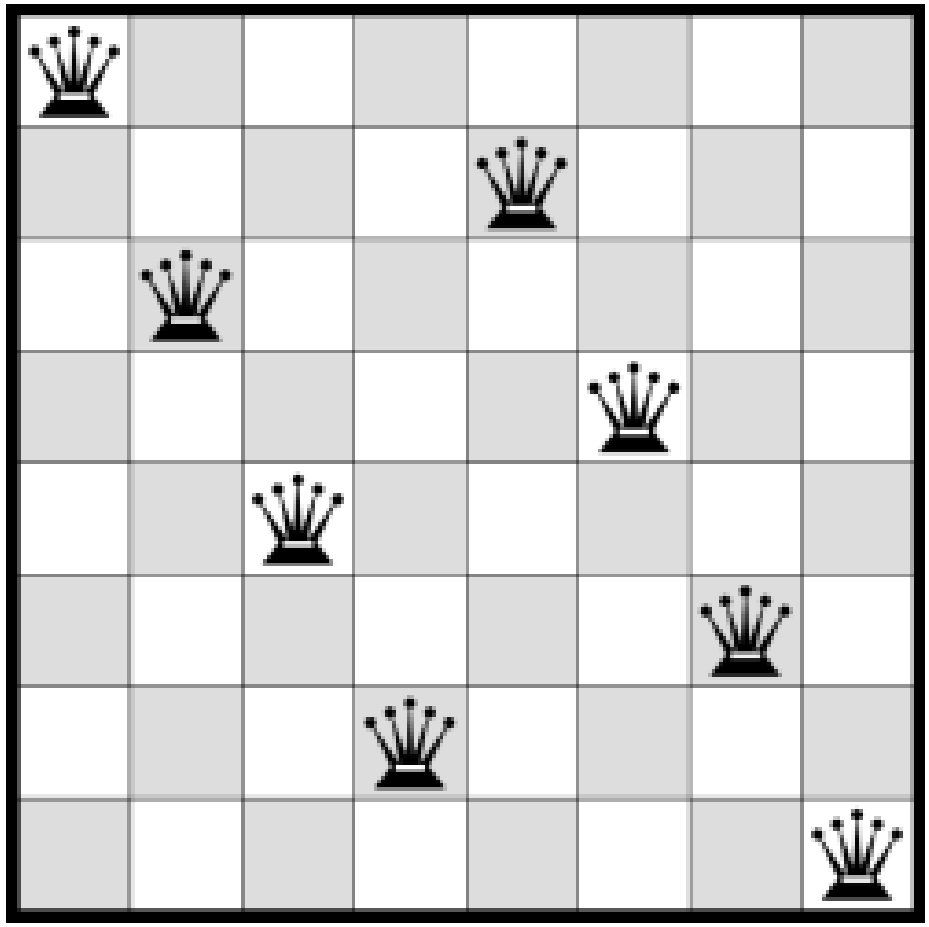
8-Puzzle Example

- Each arrangement of the numbers on the board is a state.
- There can be many initial states but there is a single goal state
- Solution: Any sequence of actions that takes us to the goal state.
- A search in the state-space is needed to find a solution.

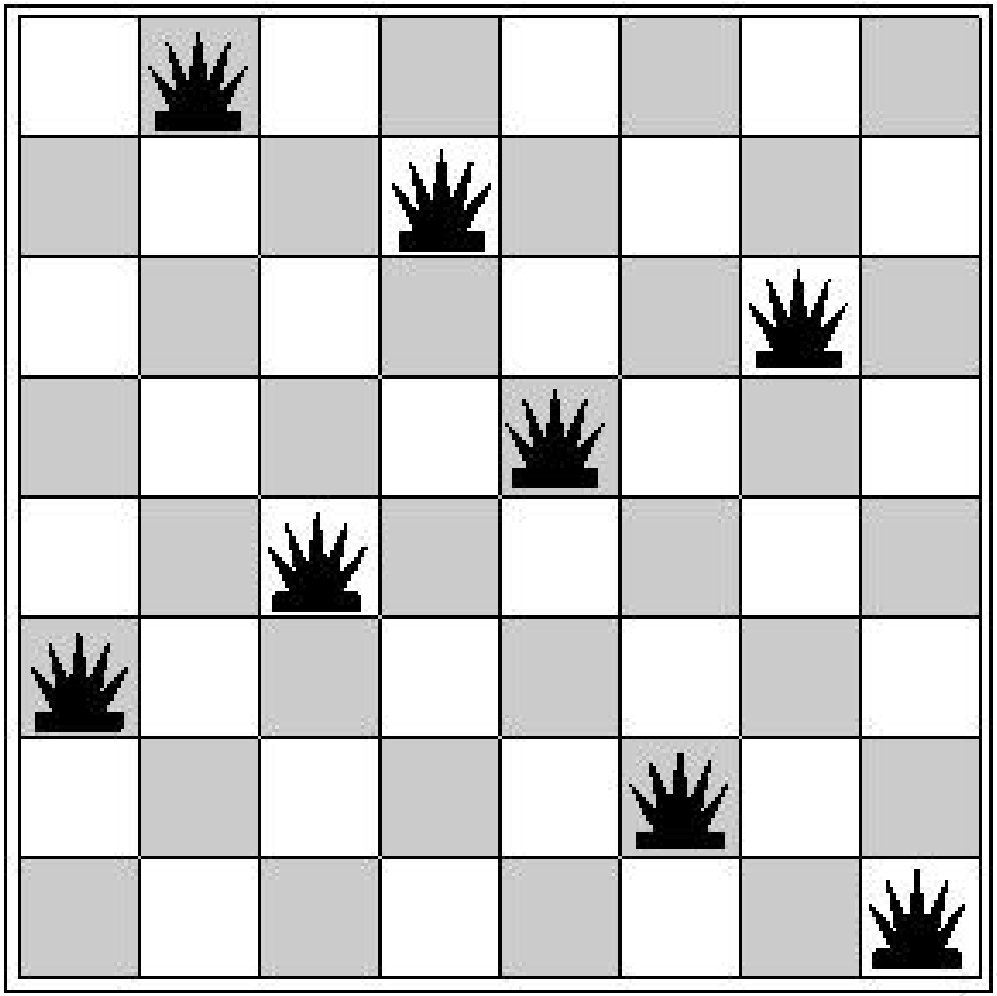
8 Queens Example

- 8 queens should be placed on a chess board so that they do not threaten each other.
- Each arrangement of the queens is a state.
- Actions: Change the position of a queen one square up/down/left/right
- Example actions: Move(Queen 3, left)

8 Queens Example: A sample state



8 Queens Problem: A goal state



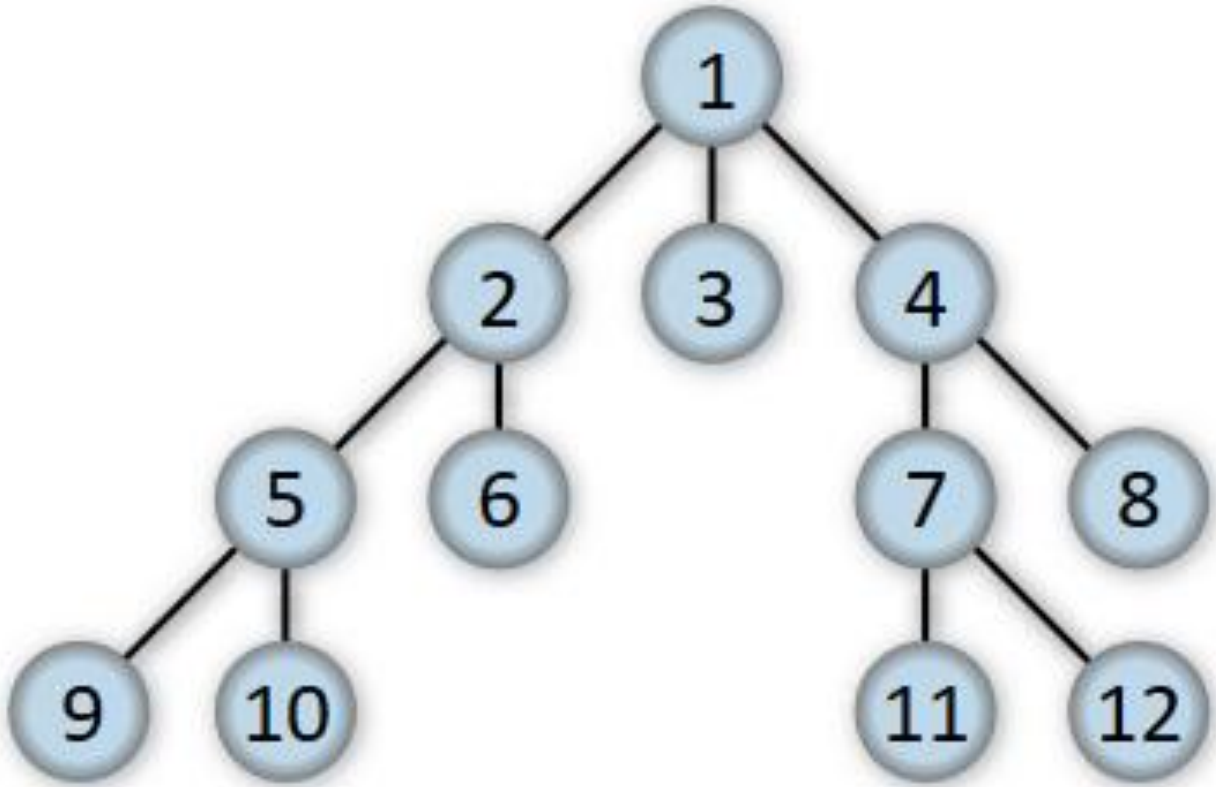
Search Categories

- 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

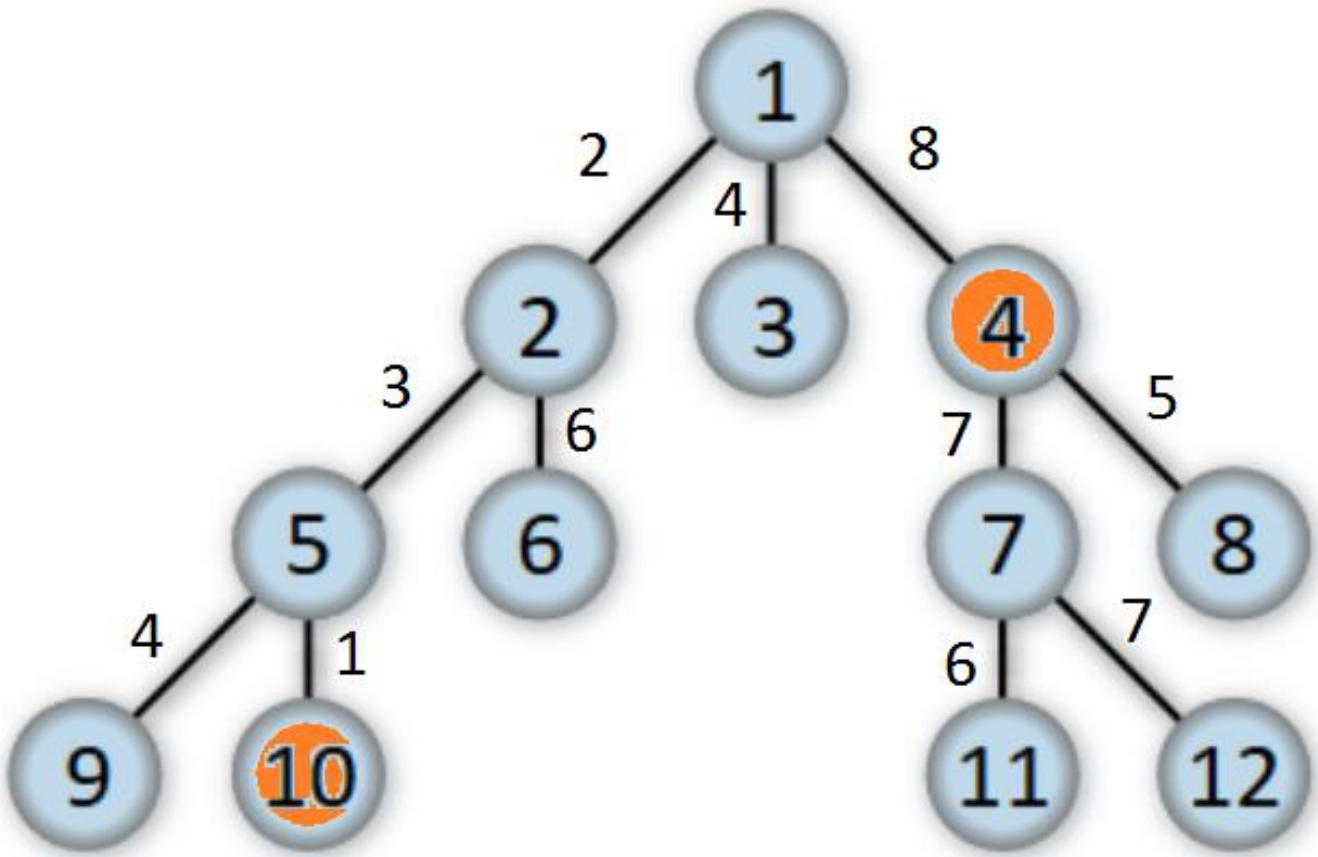
- Breadth-first search finds the *shallowest* goal state, but this may not always be the least-cost solution.

- **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.

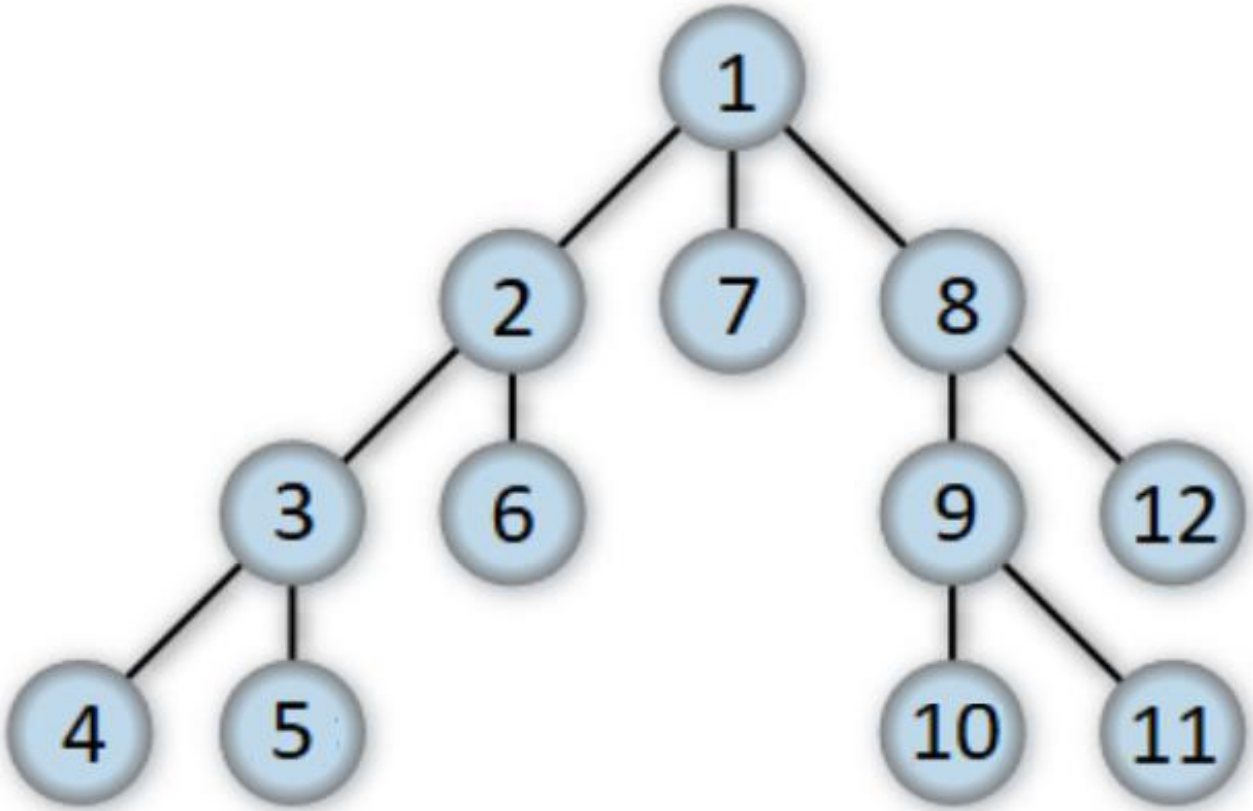
Uniform Cost Search Examples



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.

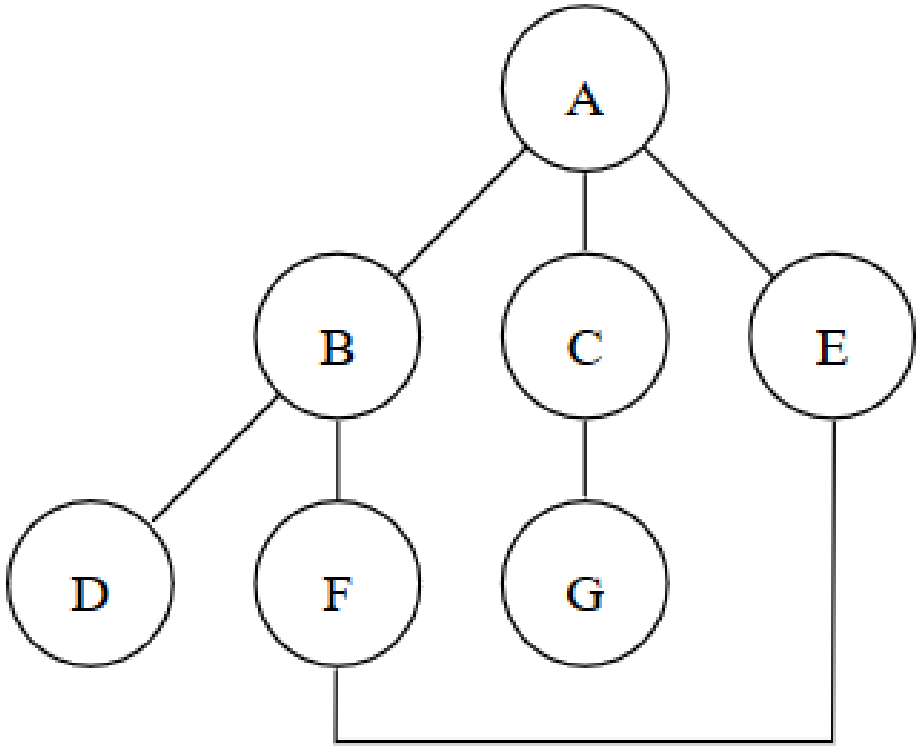
Depth First Search Example



Depth Limited Search

- **Depth-limited search** avoids the problems of depth-first search by imposing a cutoff on the maximum depth of a path.
- This cutoff can be implemented with a special depth-limited search algorithm.
- For example, when finding a path from a city to another city, if we know that there are 20 cities, so we know that the solution must be of length 19 at most.

Depth Limited Search Example



Depth Limited Search Example

- If the search algorithm can remember which states have been visited before, then starting from A we have
  - A, B, D, F, E, C, G
- Otherwise the search algorithm will be in an infinite loop:
  - A, B, D, F, E, A, B, D, F, E, etc.
- Depth limited search can solve the problem.
  - Max depth = 2 ➔ A, B, D, F, C, G, E, F

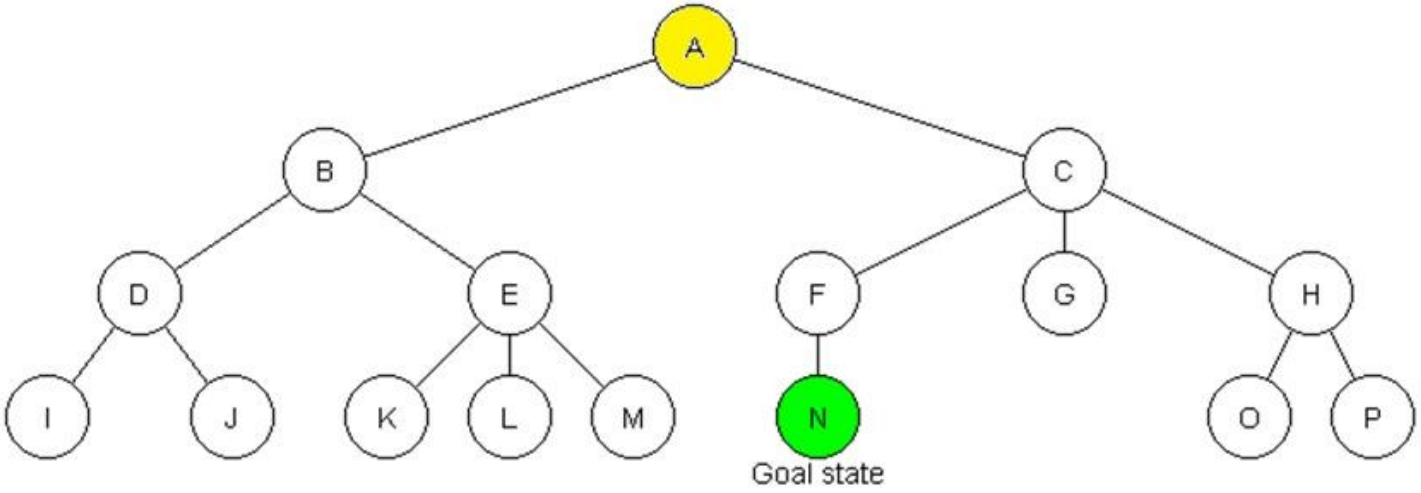
Iterative Deepening Search

- The hard part about depth-limited search is picking a good limit.
- However, for most problems, we will not know a good depth limit until we have solved the problem.
- **Iterative deepening search** starts with depth 0, then depth 1, then depth 2, and so on.
- In effect, iterative deepening combines the benefits of depth-first and breadth-first search.

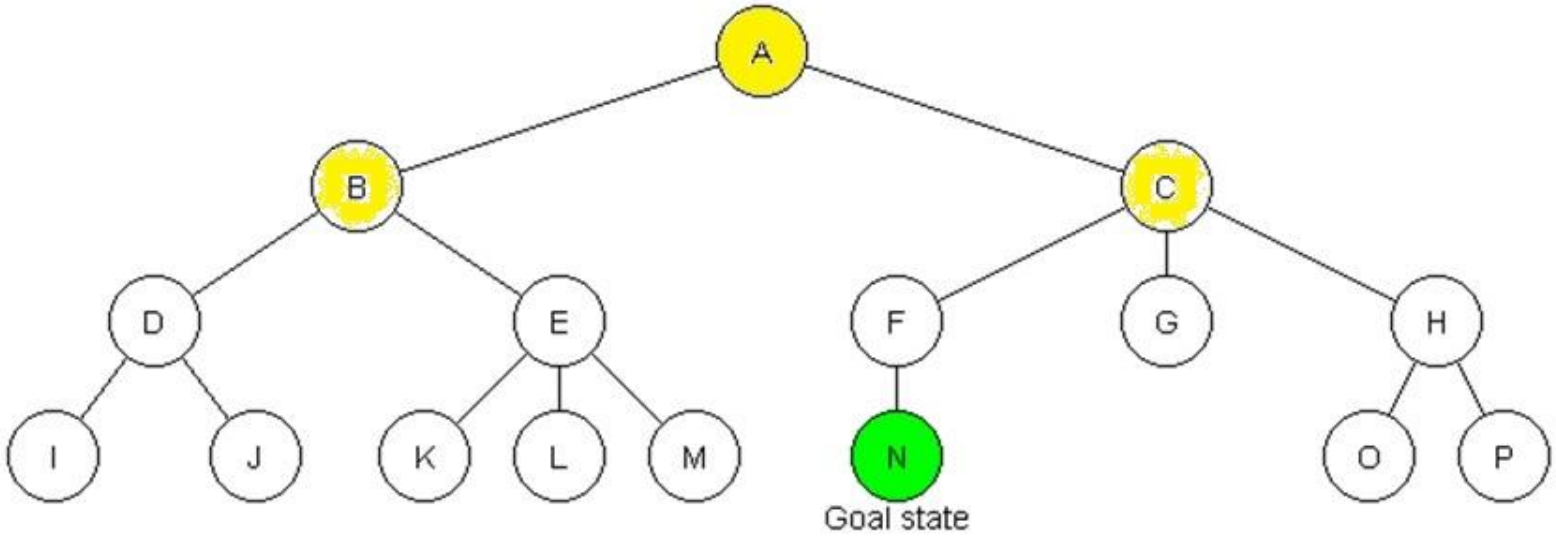
Iterative Deepening Search Example

- Assume we are looking for a goal state in chess game.
- The state-space is very large
- The search for a goal cannot be done by a depth search algorithm because the whole graph cannot be expanded

Example: Depth = 0

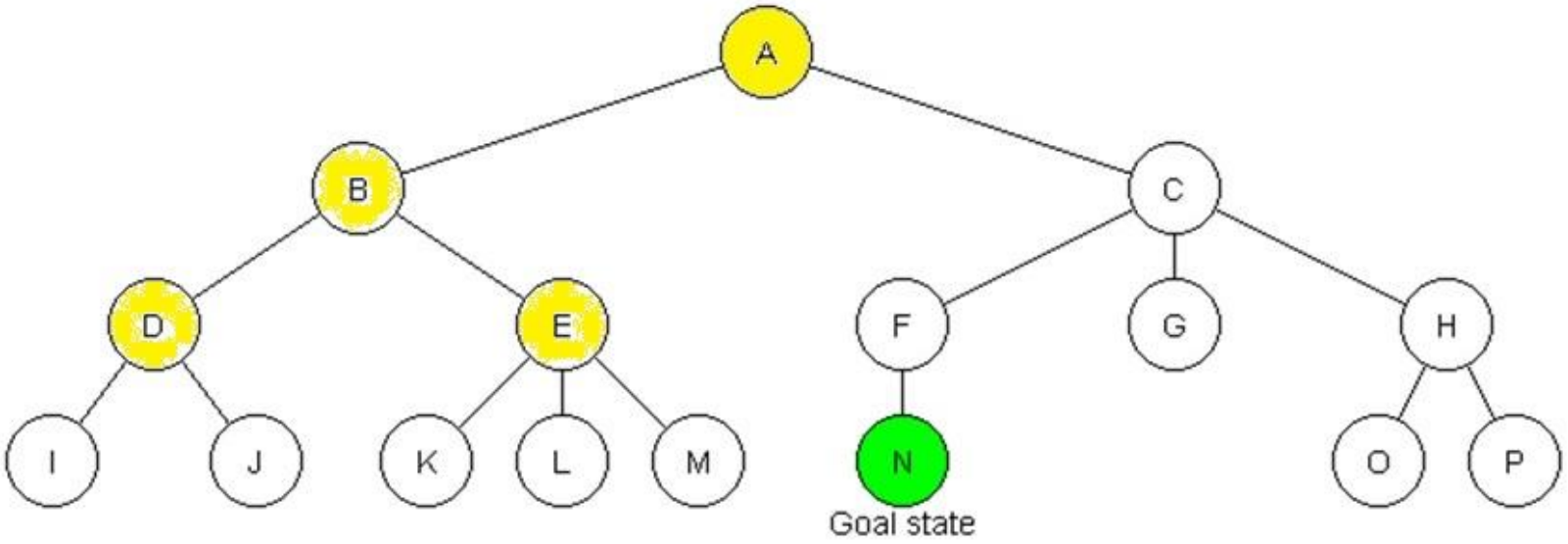


Example: Depth =1



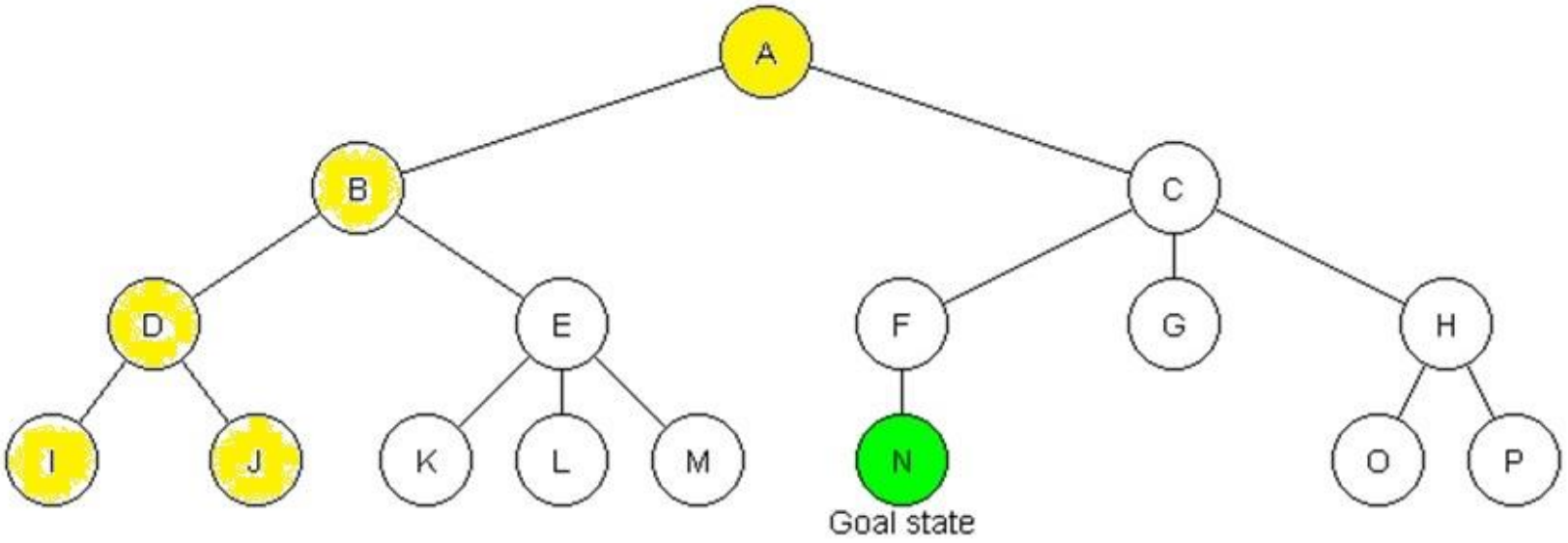
Visits: A, B, C

Example: Depth = 2



Visits: A, B, D, E, C, F, G, H

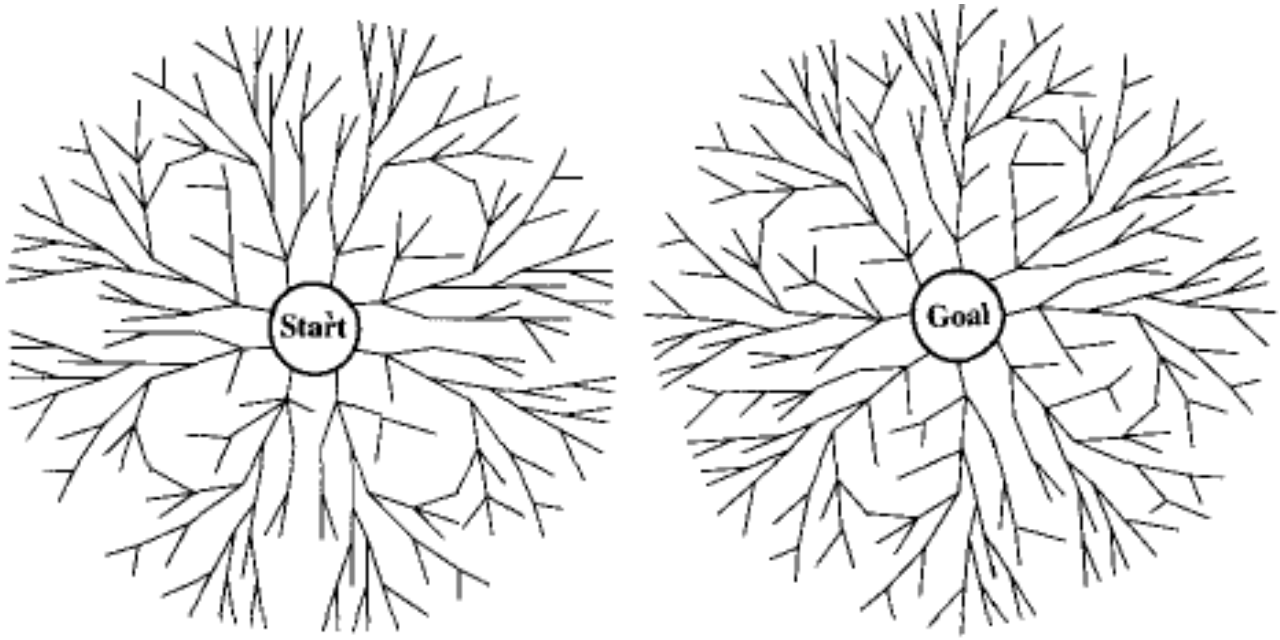
Example: Depth = 3



Visits: A, B, D, I, J, E, K, L, M C, F, N

Bidirectional Search

- The idea in bidirectional search is to search both forward from the initial state, and backward from the goal at the same time.
- The algorithm stops when the two searches meet in the middle



Constraint Satisfaction Search

- **A constraint satisfaction problem** (or CSP) is a special kind of problem that satisfies some extra conditions in addition to the basic requirements for problems.
- In a CSP, the goal test specifies a set of **constraints** that must be met.
- For example, the 8-queens problem can be viewed as a CSP with extra conditions about the locations of the queens.

Avoiding Repeating States

- There are three ways to avoid repeated states:

1. Do not return to the state that we just came from.
2. Do not create paths with cycles in them.
3. Do not generate any state that was ever generated before. This requires every state that is generated to be kept in memory.

Questions?