Uninformed Search

CS 161 Spring 2019

Sajad. Darabi Shirley. Chen

April 20, 2019

Lisp Practice

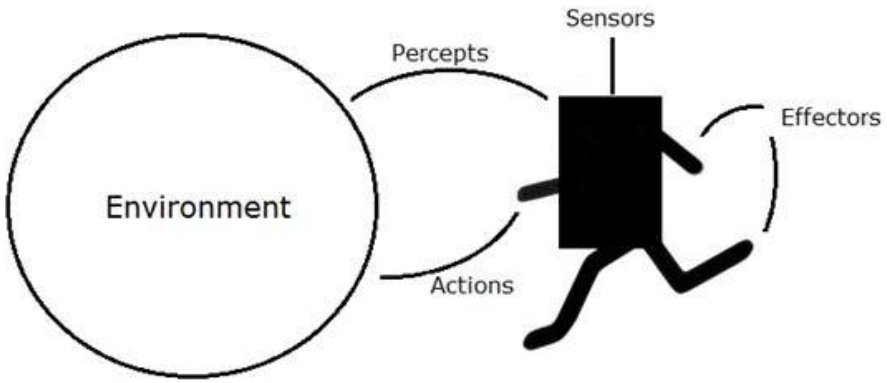
1. Write a lisp function to calculate the modulus of a number x with respect to another number y
2. Write a function to check if a list is monotonically increasing
3. Write factorial function

Disclaimer

Images included in these slides are taken from google images/slides found online. They are used for education purposes/non commercial use and follow the doctrine of **Fair Use**.

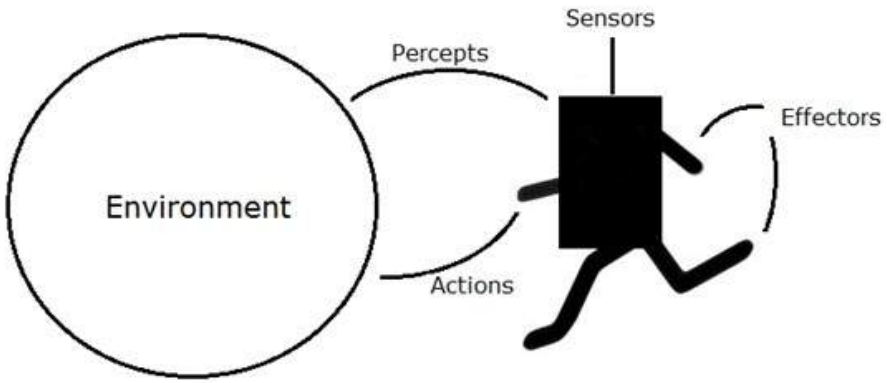
Agents

Agent interacts with the environment through various sensory inputs and acts upon them through actuators.



Agents

Agent interacts with the environment through various sensory inputs and acts upon them through actuators.



- Rational agent: they choose actions that maximize a utility cost function
- Reactive agent: choose action without planning.

Agents

Reflex agents:

- Action based on current precept of the environment
- Does not require memory of the world
- Does not consider consequence of actions

Rational agents:

- Looks ahead or hypothesizes consequence of various actions
- Must have model of the world
- Aims for an objective

Search Problem

A search problem consists of

- Initial State
- A state space $S = \{s_1, s_2, ...s_d\}$
- Successor function $F(s_t, a_t) = s_{t+1}$
- Goal test : determine if solution is achieved.

Example

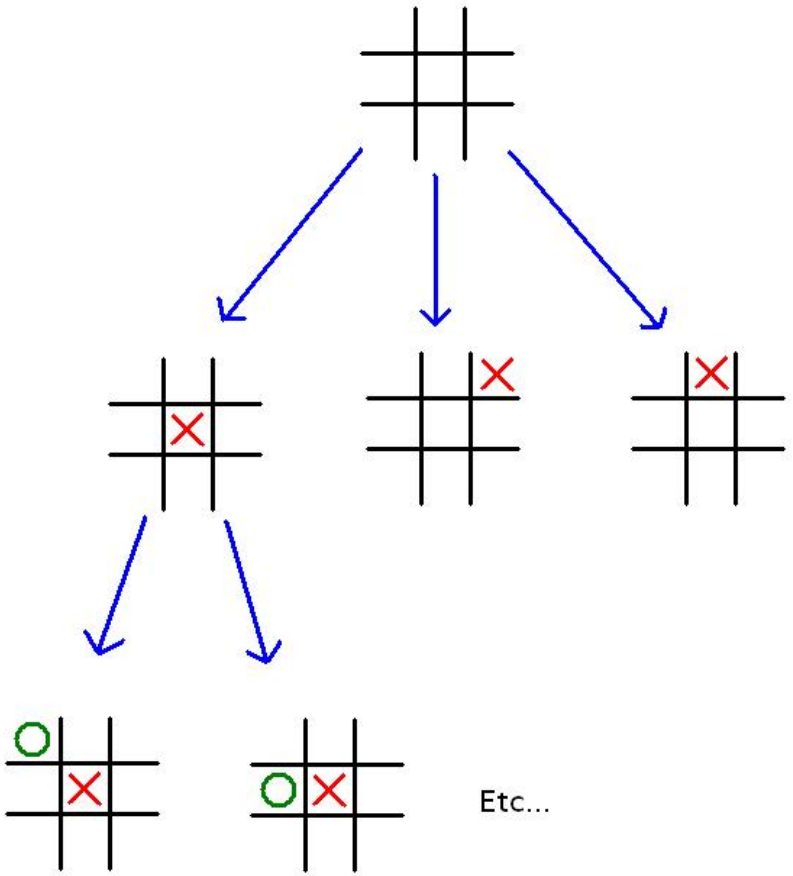
A robot arm with 3 joints J_1, J_2, J_3. All joints are initially at point 0 deg. They can be rotated by one degree steps each at a single time. The goal is to reach certain target between $0 \leq J \leq 180$. Assume the arms are rotating around a single axes.

Formulate the problem as a search problem.

Search Algorithm Evaluation

- Completeness: is it guaranteed to find the solution?
- Optimally: does it find the best one?
- Time complexity: time it takes to find the solution
- Space complexity: amount of space required to perform the search

State Space Example



min and max branching factor?

Tic Tac Toe

- What is the search problem?
- State space size?

Another Example

We have three stacks and 12 blocks of different sizes, where the blocks are stacked on top of one another. We are limited to moving a single block at a time to another of the stacks. The goal is to sort the block in the first stack.

Formulate the search problem. What are the branching factors b_{max} and b_{min}?

Another Example

We have three stacks and 12 blocks of different sizes, where the blocks are stacked on top of one another. We are limited to moving a single block at a time to another of the stacks. The goal is to sort the block in the first stack.

Formulate the search problem. What are the branching factors b_{max} and b_{min}? ***hint*** number the blocks based on size!

Searching for Solutions

Having formulated the problem we need to solve it by considering a sequence of actions.

Searching for Solutions

Having formulated the problem we need to solve it by considering a sequence of actions.

- generating: generate children's (these nodes have not been expanded yet). Generation is when a node is allocated to memory or added to the search tree.
- expanding: from the current fringe pick a node to expand. Expansion is when you generate all the childrens of the current node.

Searching for Solutions

Having formulated the problem we need to solve it by considering a sequence of actions.

- generating: generate children's (these nodes have not been expanded yet). Generation is when a node is allocated to memory or added to the search tree.
- expanding: from the current fringe pick a node to expand. Expansion is when you generate all the childrens of the current node.

Fringe (frontier): is the set of nodes that have yet to be expanded.

Searching for Solutions

Having formulated the problem we need to solve it by considering a sequence of actions.

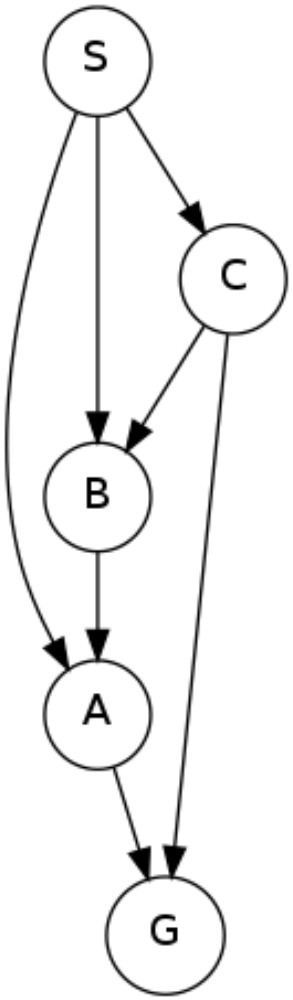
- generating: generate children's (these nodes have not been expanded yet). Generation is when a node is allocated to memory or added to the search tree.
- expanding: from the current fringe pick a node to expand. Expansion is when you generate all the childrens of the current node.

Fringe (frontier): is the set of nodes that have yet to be expanded.

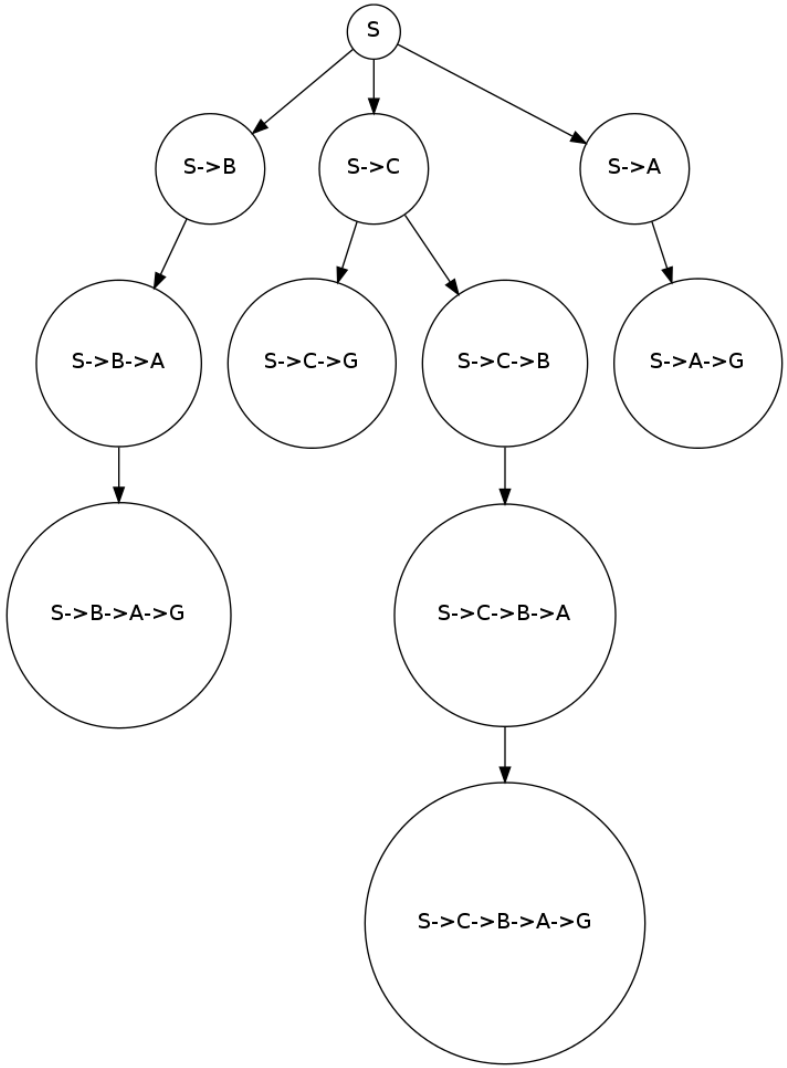
Search algorithms primarily differ on how they choose which node to expand (the search strategy).

Another example

How many nodes are in the search tree?

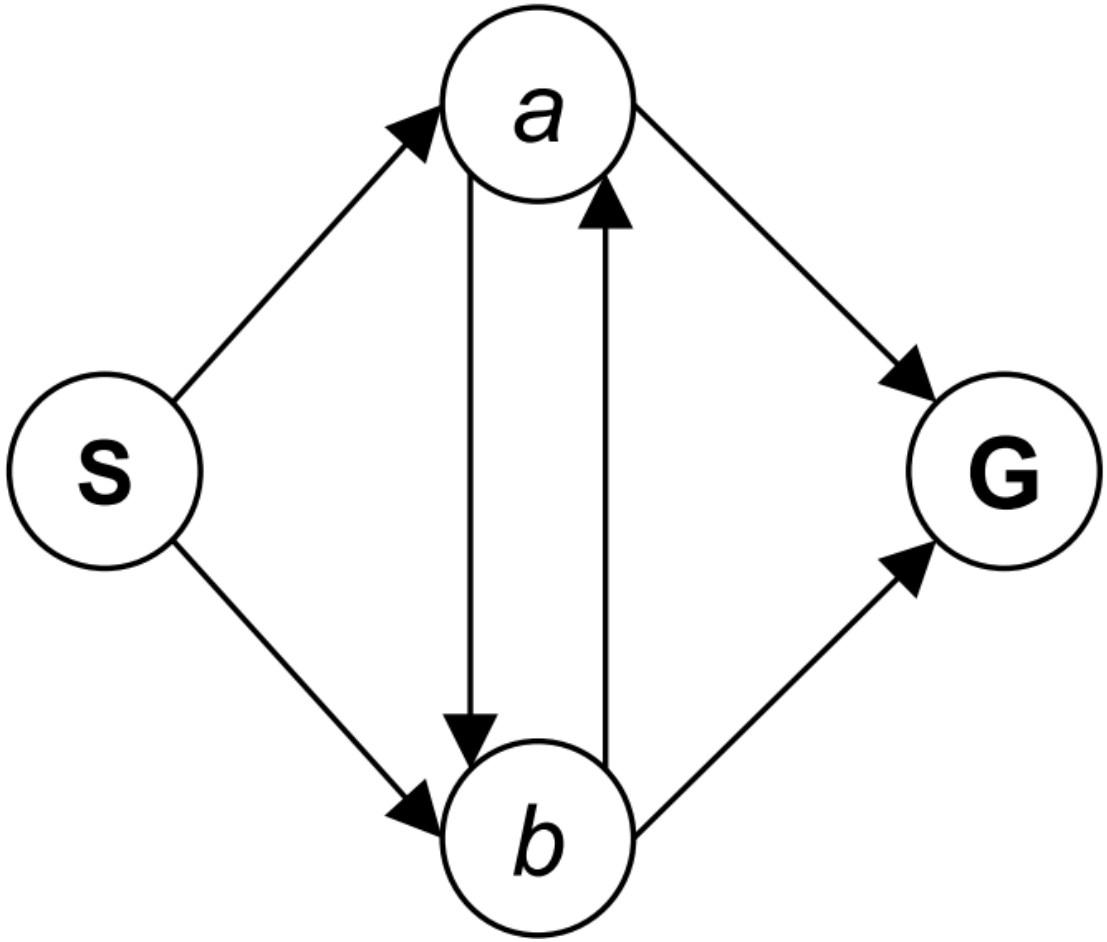


Search Tree



Search Tree versus State Space Graph

Consider the state graph below and draw its corresponding search tree.



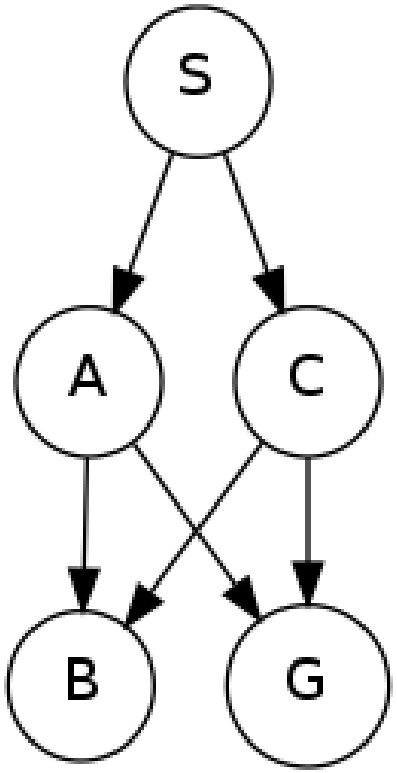
Breadth First Search (BFS)

- **Search strategy**: pick the shallowest node on the fringe to expand.
- **Goal test**: you can test for goal state upon generating new states in the fringe. (i.e. before expansion)

Note: In BFS it's possible to test on GENERATION, though in books you might have testing for goal on EXPANSION.

BFS Example

Determine how the goal is reached using depth first search.



BFS

BFS can be implemented using a queue structure (FIFO).

BFS

BFS can be implemented using a queue structure (FIFO).

- It is complete

BFS

BFS can be implemented using a queue structure (FIFO).

- It is complete
- Optimal (if the costs are uniform or non-decreasing function)

BFS

BFS can be implemented using a queue structure (FIFO).

- It is complete
- Optimal (if the costs are uniform or non-decreasing function)
- Memory Time $O(b^d)$
  - b: branching factor
  - d: depth of the solution
  - m: tree height

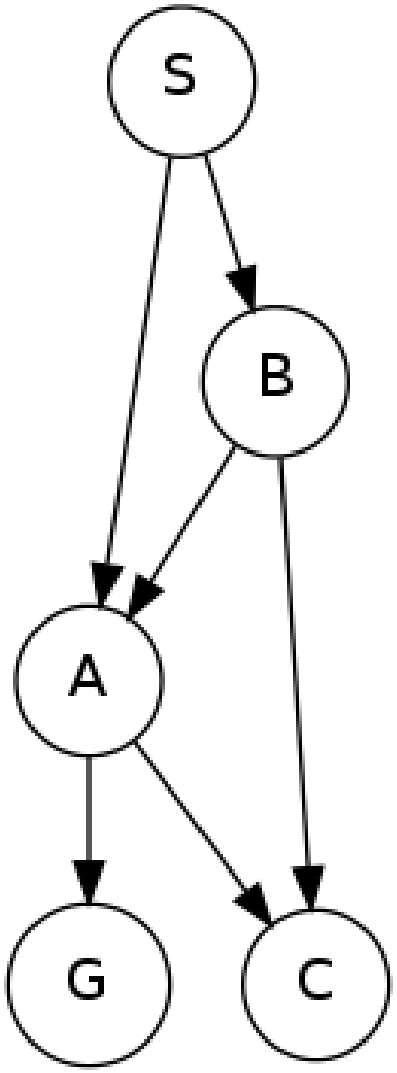
Depth First Search

Search strategy: pick the deepest node on the fringe to expand.

Goal test: can test for goal state upon expanding nodes.

DFS Example

Determine how the goal is reached using depth first search.



DFS

Depth first search can be implemented using a stack (FILO)

- Incomplete

DFS

Depth first search can be implemented using a stack (FILO)

- Incomplete
- Not Optimal

DFS

Depth first search can be implemented using a stack (FILO)

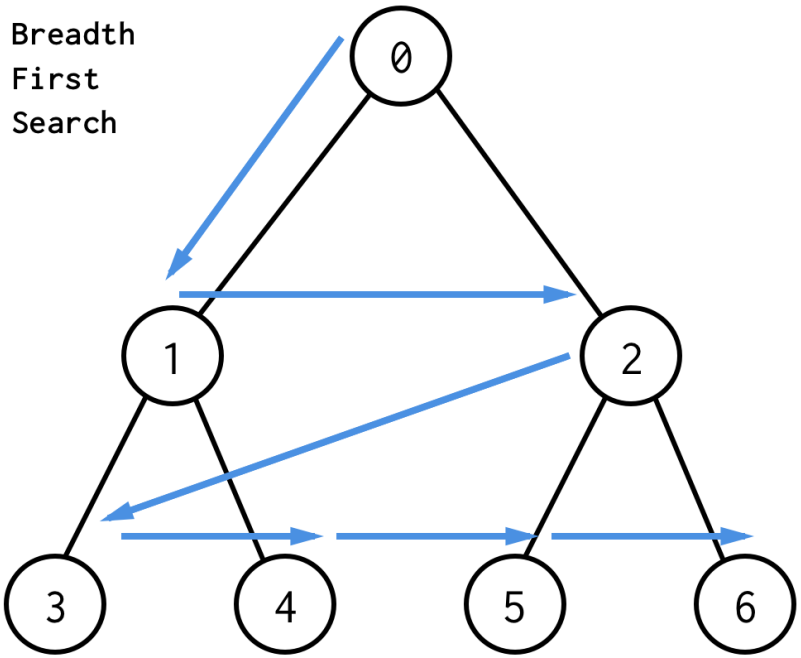
- Incomplete
- Not Optimal
- Time $O(b^m)$

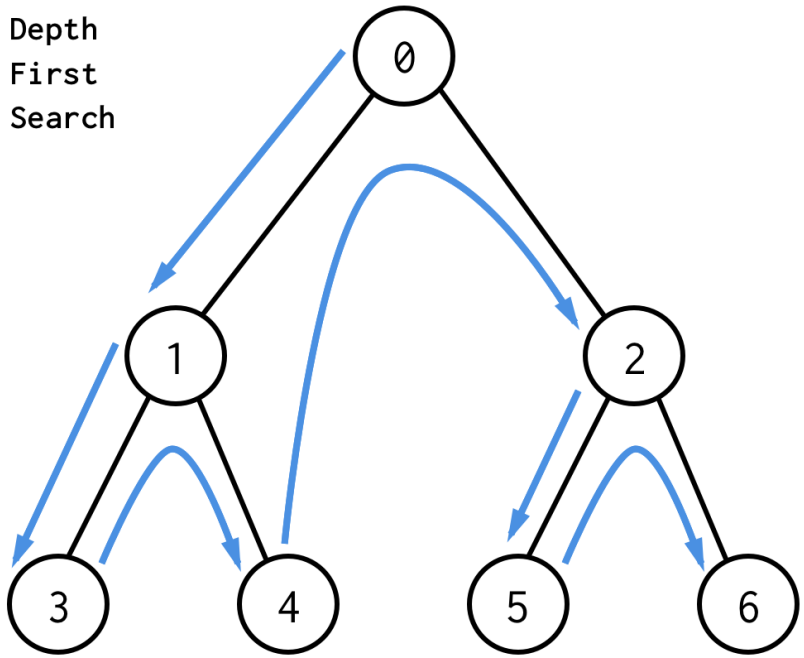
DFS

Depth first search can be implemented using a stack (FILO)

- Incomplete
- Not Optimal
- Time $O(b^m)$
- Space $O(bm)$

DFS vs BFS





Thank You!