Reminder: We do have class tomorrow. K-day starts at noon.

Chapter 3 Solving Problems by Searching

Section 3.1 - 3.4 Uninformed Search Strategies

a problem

\[ \text{states} \]

\[ \text{a snapshot of the world} \]

- initial state only one
- goal state(s)

method GOAL-TEST(state)

\[ \text{actions} \]

- action-name(parameters)

- goal state

Go (Hagston-downtown)

Not a method but descriptive text for the action

transition model
(describes how an action transitions the state from one state to another)

RESULT(state, action)

method returns the next state

RESULT(at(MITU-campus), go (Hagston-downtown))

at (Hagston-downtown)
Cell SEARCH (initial state, goal test(s), actions, transition model)

action library implementation of RESULT

Construct a "search tree"

not a graph
(nodes are repeated by duplicating)
Example search problem

Figure 3.2 A simplified road map of part of Romania.

search for a route from Arad to Bucharest

initial state

goal state

action: go

build the search tree

root node: it contains the initial state

children
order is arbitrary

Arad

Zerind

Orașel

Arad

Bucharest

Rimnicu Vilea

Sibiu

Fagaras

Urziceni

Vaslui

Iași

Giurgiu

Hirsova

Eforie

n: cities

map: represent as a graph

matrix: adjacency list
frontier
list of search nodes

depth-first

breadth-first

"expand" a state
find its children
(finding all possible
actions from that
state and where
they lead to)

frontier

OAST
put the new children
in to the front
of the frontier

put the new children
to the back of the
frontier
Why would we choose between DFS and BFS:

- Possibility of loops with DFS
- Takes time to reach the states on the "right" exploration strategy
- Memory requirements

Systematic search

No advantage to doing DFS, BFS

Compare search strategies:

1. Completeness
   - If there is a path from initial to goal, is the search guaranteed to find it?

2. Time complexity
   - How much time do we need in the worst case to find the goal?

3. Space complexity
   - How much storage do we need during the search?

4. Optimality
   - What is the cost of the path, is it the "best" cost?
How to measure

1. completeness
   proof that it is true

2. time complexity
   "measure" the time spent
   count the number of nodes visited (expanded)
   OR
   count the number of nodes generated
   \( n \times \text{processing time} \)
   exponential time complexity