Artificial Intelligence

Jeff Clune
Assistant Professor
Evolving Artificial Intelligence Laboratory

University of Wyoming
The Gambler

Every gambler knows that the secret to survivin'
Is knowin' what to throw away and knowing what to keep
'Cause every hand's a winner and every hand's a loser
The Gambler

Every gambler knows that the secret to survivin'
Is knowin' what to throw away and knowing what to keep
'Cause every hand's a winner and every hand's a loser
And the best that you can hope for is to die in your sleep
to maximize expected reap!
Search

• Huge, important category of AI
• Applies to many domains
Searching Algorithms

systematic ways of looking for things

cheese in a maze  

lost keys in a house
Searching Algorithms

systematic ways of looking for things

social networks: how many degrees of separation to kevin bacon?

driving directions (path planning)
Graph Representation
Graph Representation

moving in a house

(humans or robots)
Graph Representation

Networks
- social
- highway
- biological
  - gene, neural, protein, etc.
- communication
- many more...

social

highway
Tree Representation

(connected graphs with no cycles)
Search Tree

- Nodes: states
- Edges: actions
State Space

- All possible states
  - (reachable from the starting point via a sequence of actions)
Path

- One sequence of actions through state space
Step Cost & Path Cost

• Expense of traversing the step/path
  • possible step/path costs for driving from Arad to Bucharest?
  • Note: assume step costs are non-negative
Algorithm

- Goal: an algorithm that takes a problem (properly described) and returns the optimal solution
  - the solution with the lowest path cost
  - the book says what “properly described” means in this context
    - and provides examples
Tree Search

```plaintext
function TREE-SEARCH(problem) returns a solution, or failure
    initialize the frontier using the initial state of problem
    loop do
        if the frontier is empty then return failure
        choose a leaf node and remove it from the frontier
        if the node contains a goal state then return the corresponding solution
        expand the chosen node, adding the resulting nodes to the frontier
```

In pairs: try it and record the order of nodes visited (solution/goal = 19, then do it again for 67)

Expand nodes in the order added (FIFO).

Add nodes clockwise (start at midnight)
Tree Search

function TREE-SEARCH(problem) returns a solution, or failure
initialize the frontier using the initial state of problem
loop do
    if the frontier is empty then return failure
    choose a leaf node and remove it from the frontier
    if the node contains a goal state then return the corresponding solution
    expand the chosen node, adding the resulting nodes to the frontier

In pairs: try it and record the order of nodes visited (goal=bucaharest)

Expand nodes in the order added (FIFO).

Add nodes clockwise (start at midnight)