Informed search algorithms

Chapter 3

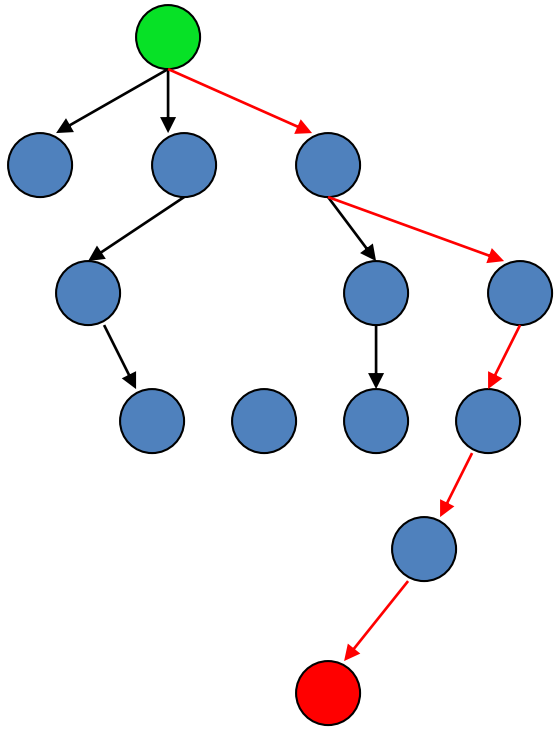
(Based on Slides by Stuart Russell, Richard Korf, Subbarao Kambhampati, and UW-AI faculty)

“Intuition, like the rays of the sun, acts only in an inflexibly straight line; it can guess right only on condition of never diverting its gaze; the freaks of chance disturb it.”

-- Honore de Balzac

Informed (Heuristic) Search

Idea: be **smart** about what paths to try.



Blind Search vs. Informed Search

- What's the difference?

- How do we formally specify this?

  A node is selected for expansion based on an evaluation function that estimates cost to goal.

General Tree Search Paradigm

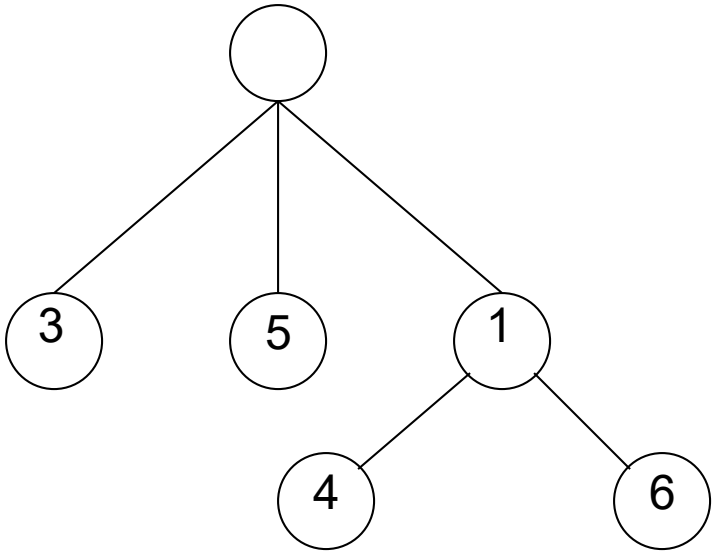
```
function tree-search(root-node)
   fringe ← successors(root-node)
   while ( notempty(fringe) )
         {node ← remove-first(fringe) //lowest f value
            state ← state(node)
            if goal-test(state) return solution(node)
            fringe ← insert-all(successors(node),fringe) }
   return failure
end tree-search
```

General Graph Search Paradigm

```
function tree-search(root-node)
   fringe ← successors(root-node)
   explored ← empty
   while ( notempty(fringe) )
         {node ← remove-first(fringe)
             state ← state(node)
             if goal-test(state) return solution(node)
             explored ← insert(node,explored)
             fringe ← insert-all(successors(node),fringe, if node not in explored)
         }
   return failure
end tree-search
```

Best-First Search

- Use an evaluation function f(n) for node n.
- Always choose the node from fringe that has the lowest f value.

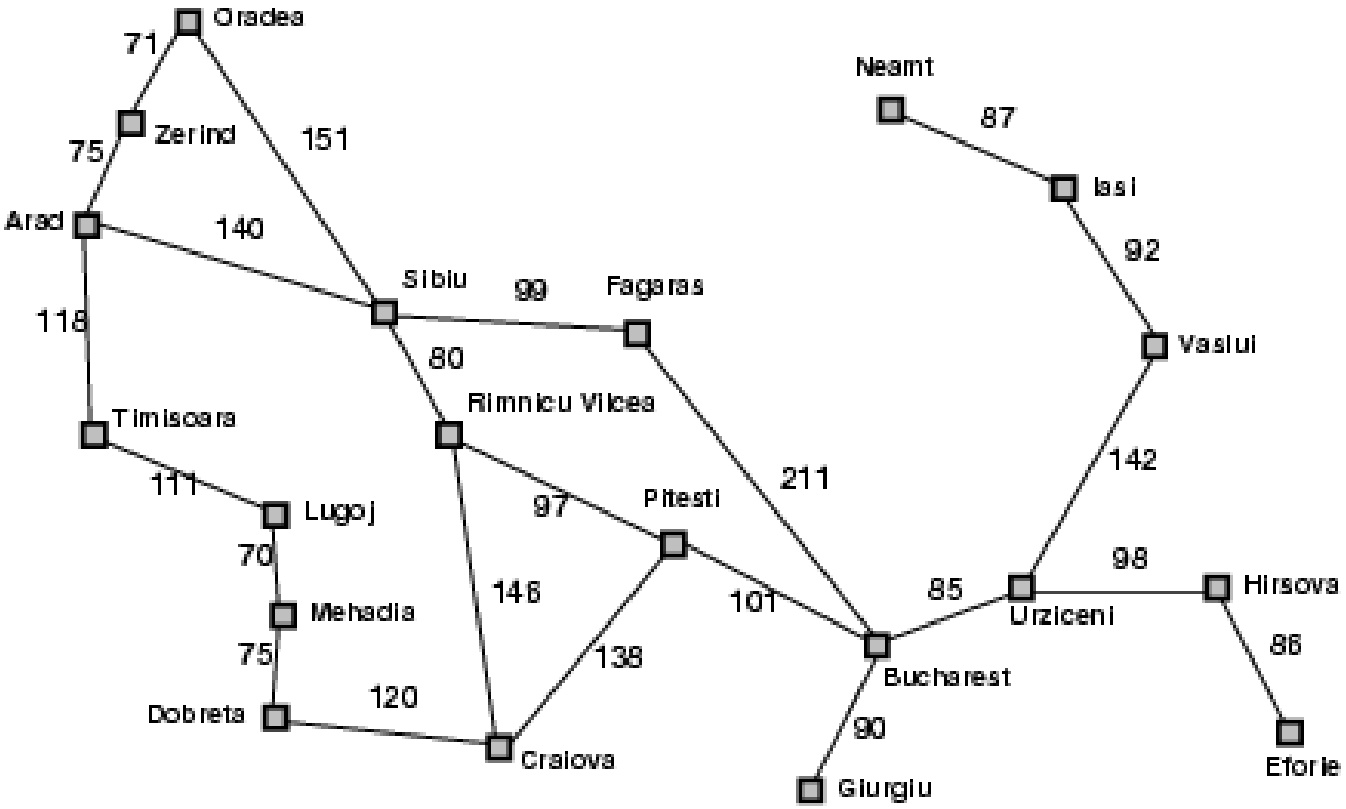


Best-first search

- A search strategy is defined by picking the order of node expansion
- Idea: use an evaluation function *f(n)* for each node
  - estimate of "desirability“

  → Expand most desirable unexpanded node
- Implementation:
  Order the nodes in fringe in decreasing order of desirability
- Special cases:
  - greedy best-first search
  - A* search

Romania with step costs in km



Old (Uninformed) Friends

- Breadth First =
  - Best First
  - with f(n) = depth(n)

- Uniform cost search =
  - Best First
  - with f(n) = the sum of edge costs from start to n

g(n)

Greedy best-first search

- Evaluation function $f(n) = h(n)$ (heuristic function)
  = estimate of cost from n to *goal*

- e.g., $h_{SLD}(n)$ = straight-line distance from n to Bucharest

- Greedy best-first search expands the node that appears to be closest to goal

Properties of greedy best-first search

- Complete?
  - No – can get stuck in loops, e.g., Iasi → Neamt → Iasi → Neamt →
- Time?
  - $O(b^m)$, but a good heuristic can give dramatic improvement
- Space?
  - $O(b^m)$ -- keeps all nodes in memory
- Optimal?
  - No