ARTIFICIAL INTELLIGENCE

Russell & Norvig

Chapter 3: Solving Problems by Searching, part 3

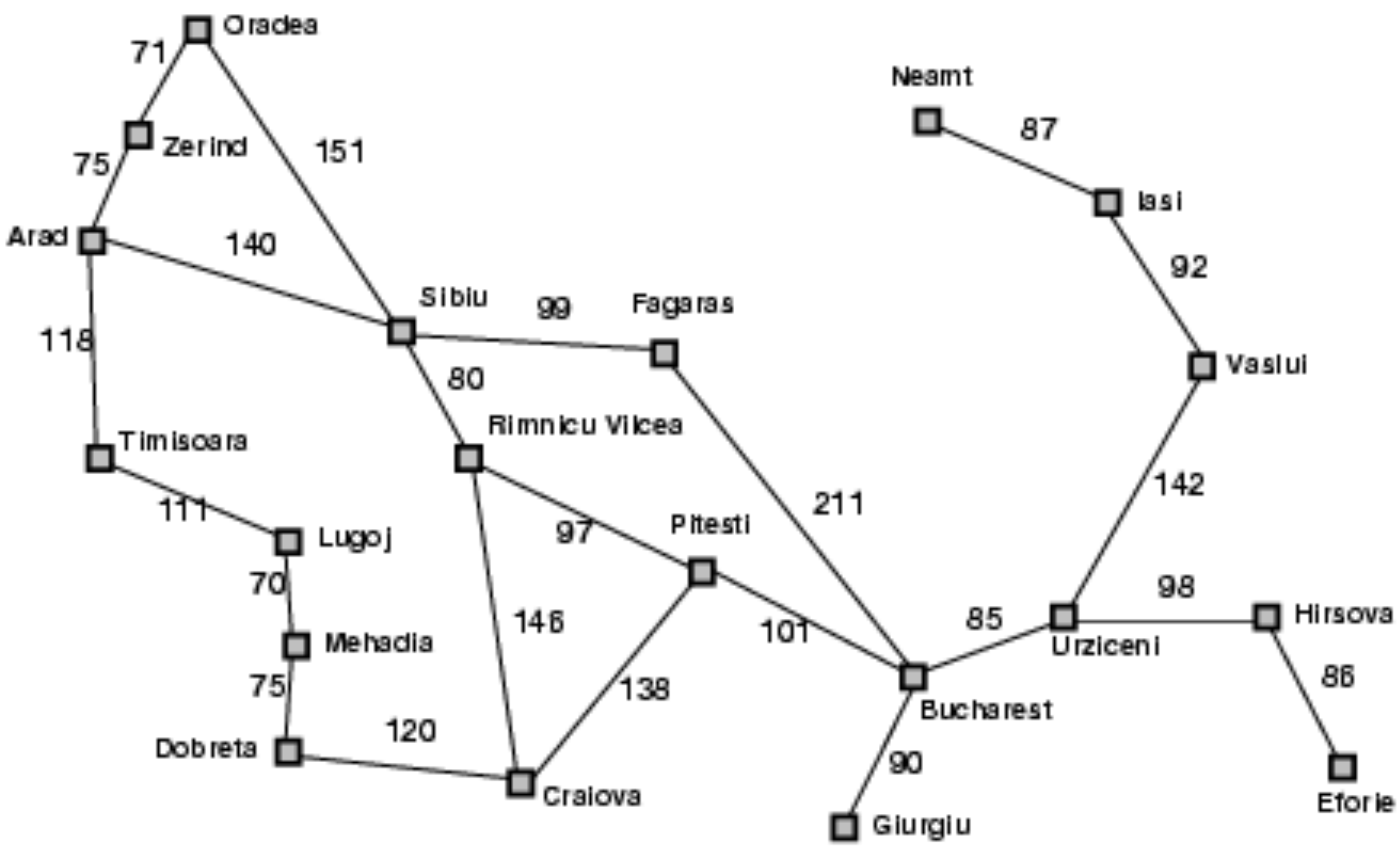
Informed (heuristic) search

- Greedy best-first search
- A* search
  - Optimality
  - Admissibility
  - Consistency
  - Controlling memory needs
- Heuristic functions

Informed search

- Uses problem-specific knowledge beyond the definition of the problem itself
- Finds solutions more efficiently than uninformed searches
- Greedy best-first
  - Heuristic h(n) is estimated cost of the cheapest path from node n to a goal state
  - Greedy best-first uses h(n) to choose next node to expand
  - Consider straight-line distance in Romania

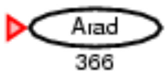
Romania with SLD



Straight–line distance to Bucharest

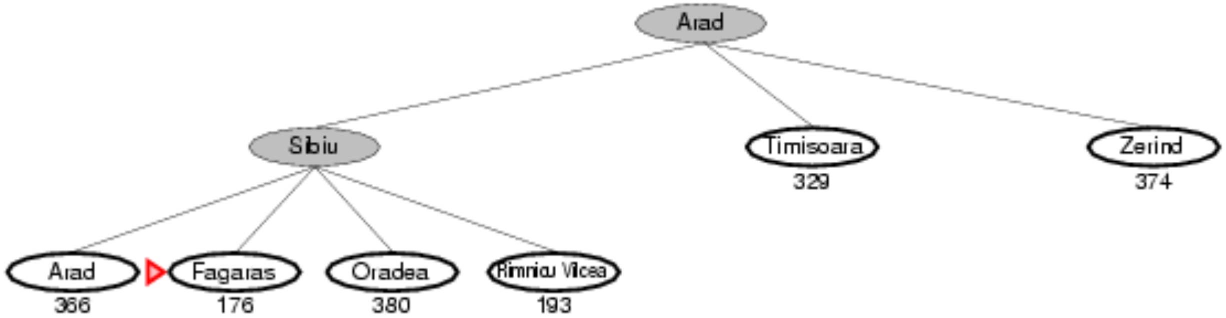
Arad	366
Bucharest	0
Craiova	160
Dobreta	242
Eforie	161
Fagaras	176
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadia	241
Neamt	234
Oradea	380
Pitesti	10
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

Greedy best-first search example

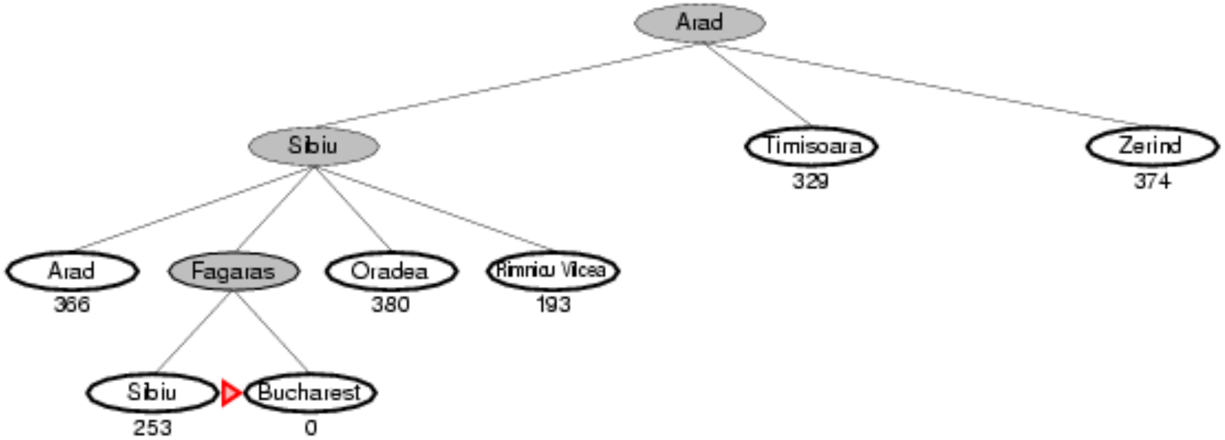


Greedy best-first search example

Greedy best-first search example



Greedy best-first search example



Problems with best-first

- Greedy best-first search is not optimal
  - Path found in Romania was Arad➔Sibiu➔Fagaras➔Bucharest, which has path cost 140+99+211 = 450 km
  - Better path would be Arad➔Sibiu➔RV➔Pitesti➔Bucharest, which has path cost 140+80+97+101 = 418 km (32 km closer)

- It is also incomplete (consider searching for path from Iasi to Fagaras) in tree-based search

- Worst-case time and space complexity is $O(b^m)$ where m is the maximum depth of the search space
- Actual performance depends on quality of heuristic

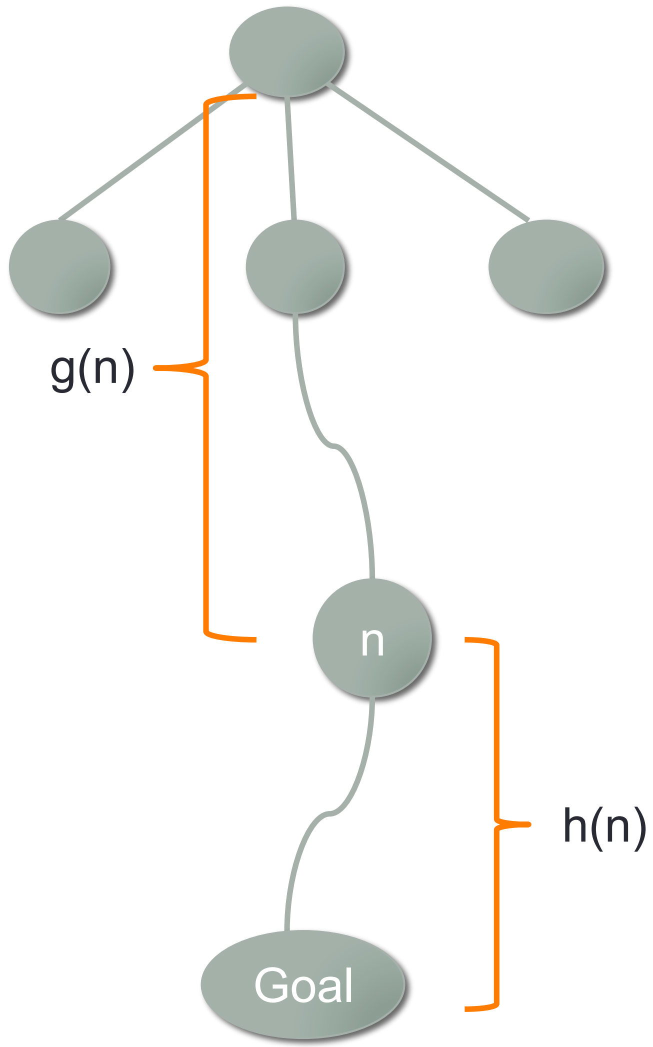
A* search (common best-first search)

- Uses evaluation function

  $f(n)=g(n)+h(n)$

- g(n) is path cost to node n
- h(n) is estimated cheapest path from node n to a goal

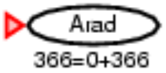
- f(n) is estimated cost of cheapest solution through node n



More A*

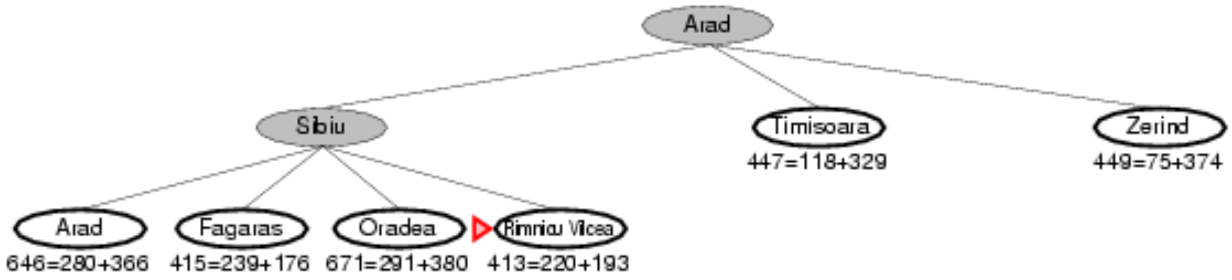
- Identical to Uniform cost search except A* uses g+h instead of g
- When heuristic function h(n) satisfies some conditions, A* is both complete and optimal.

A* search example

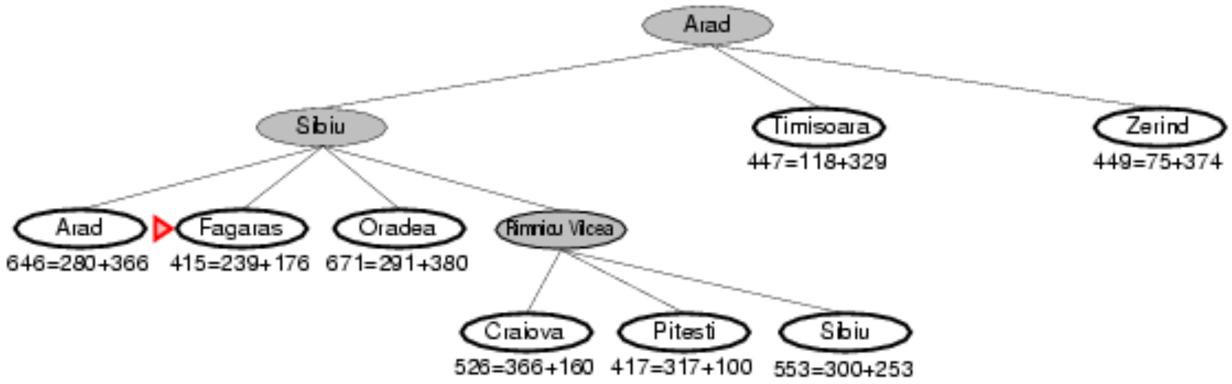


A* search example

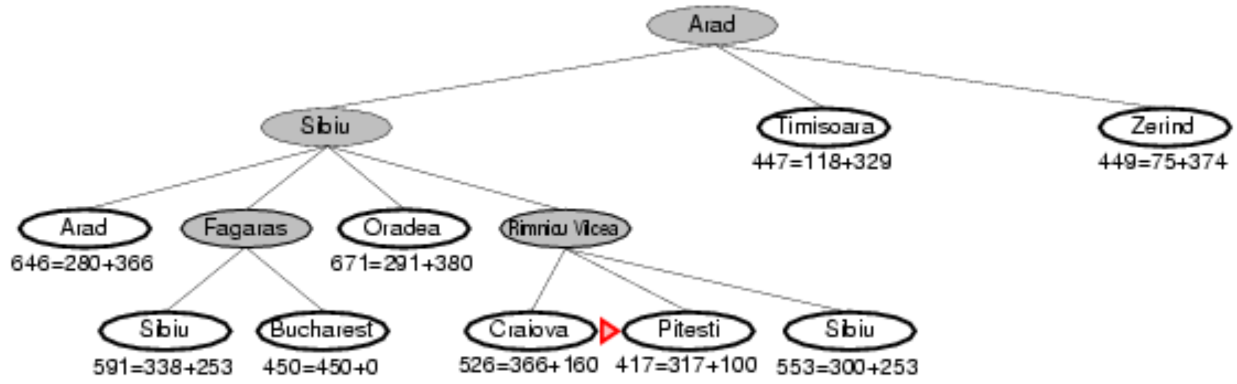
A* search example



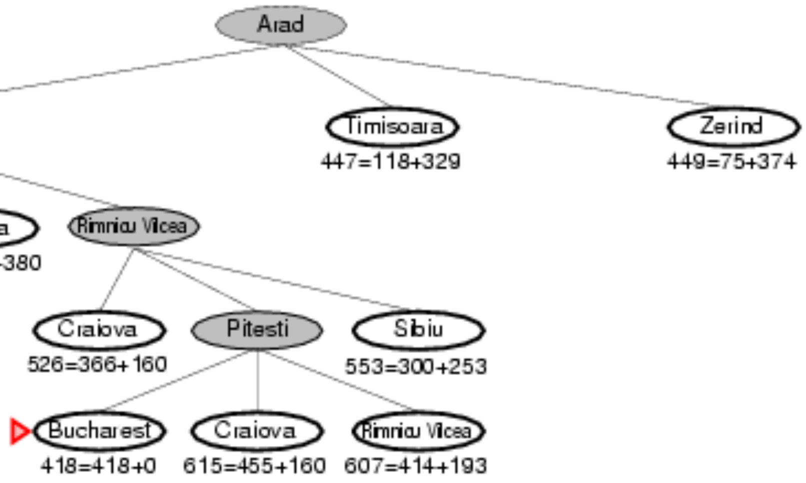
A* search example



A* search example



A* search example

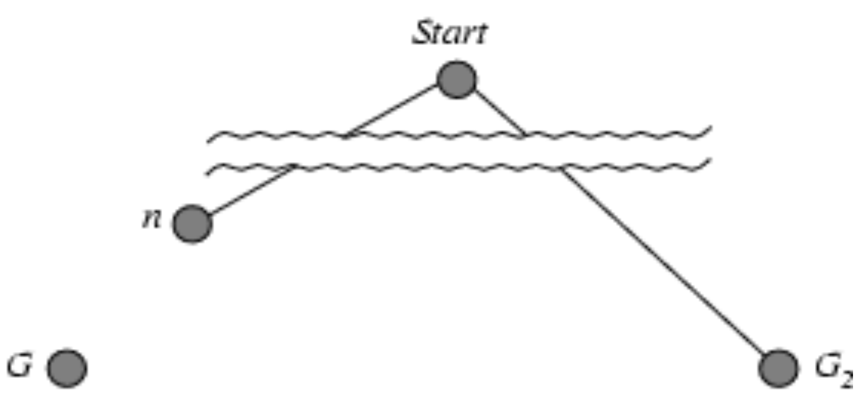


Admissible heuristics

- A heuristic *h(n)* is **admissible** if for every node *n*, $h(n) \leq h^*(n)$, where $h^*(n)$ is the *true* cost to reach the goal state from *n*.
- An admissible heuristic *never overestimates* the cost to reach the goal, i.e., it is **optimistic**
- Example: $h_{SLD}(n)$ (never overestimates the actual road distance)
- Theorem: If *h(n)* is admissible, A^* using `TREE-SEARCH` is optimal

Optimality of A^* (proof)

- Suppose some suboptimal goal G_2 has been generated and is in the fringe. Let *n* be an unexpanded node in the fringe such that *n* is on a shortest path to an optimal goal *G*.



- $f(G_2) = g(G_2)$ since $h(G_2) = 0$
- $g(G_2) > g(G)$ since G_2 is suboptimal
- $f(G) = g(G)$ since $h(G) = 0$
- $f(G_2) > f(G)$ from above
- $h(n) \leq h^*(n)$ since h is admissible

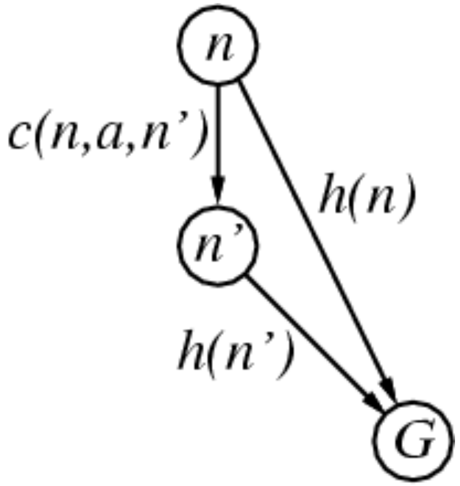
- $g(n) + h(n) \leq g(n) + h^*(n)$
- $f(n) \leq f(G)$

Hence $f(G_2) > f(n)$, and A^* will never select G_2 for expansion

Consistent heuristics

- A heuristic is **consistent** if for every node *n*, every successor *n'* of *n* generated by any action *a*,

$h(n) \leq c(n,a,n') + h(n')$



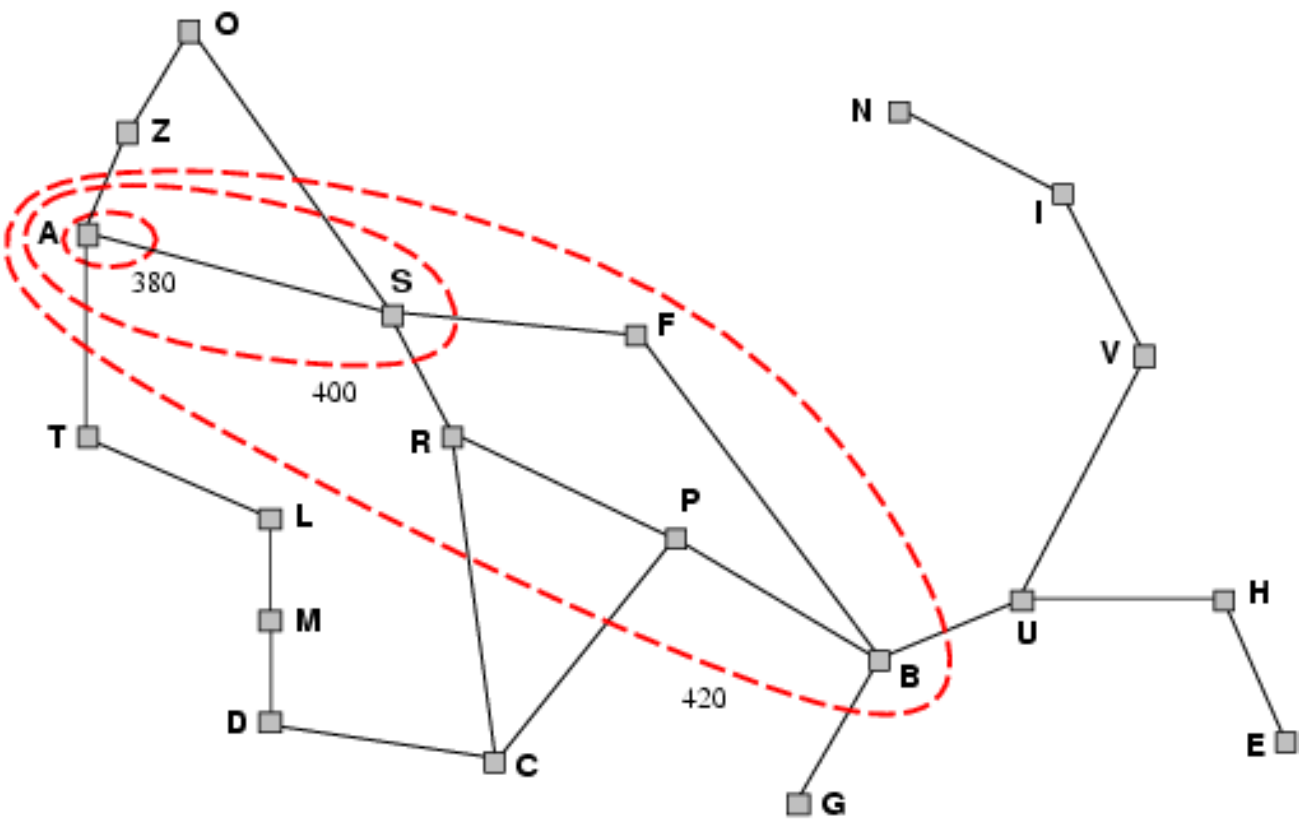
- If *h* is consistent, we have

$$
\begin{aligned}
f(n') &= g(n') + h(n') \\
&= g(n) + c(n,a,n') + h(n') \\
&\geq g(n) + h(n) \\
&= f(n)
\end{aligned}
$$

- i.e., *f(n)* is non-decreasing along any path.
- Theorem: If *h(n)* is consistent, A* using `GRAPH-SEARCH` is optimal

Optimality of A*

- A* expands nodes in order of increasing *f* value
- Gradually adds "*f*-contours" of nodes
- Contour *i* has all nodes with $f=f_i$, where $f_i < f_{i+1}$



Properties of A*

- It is **complete**, unless there are infinitely many nodes with f ≤ *f(G)*
- It is **optimal**.
- It is **optimally efficient**, that is, no other optimal algorithm is guaranteed to expand fewer nodes than A* (except on tie-break among nodes with f(n)=cost of optimal path)

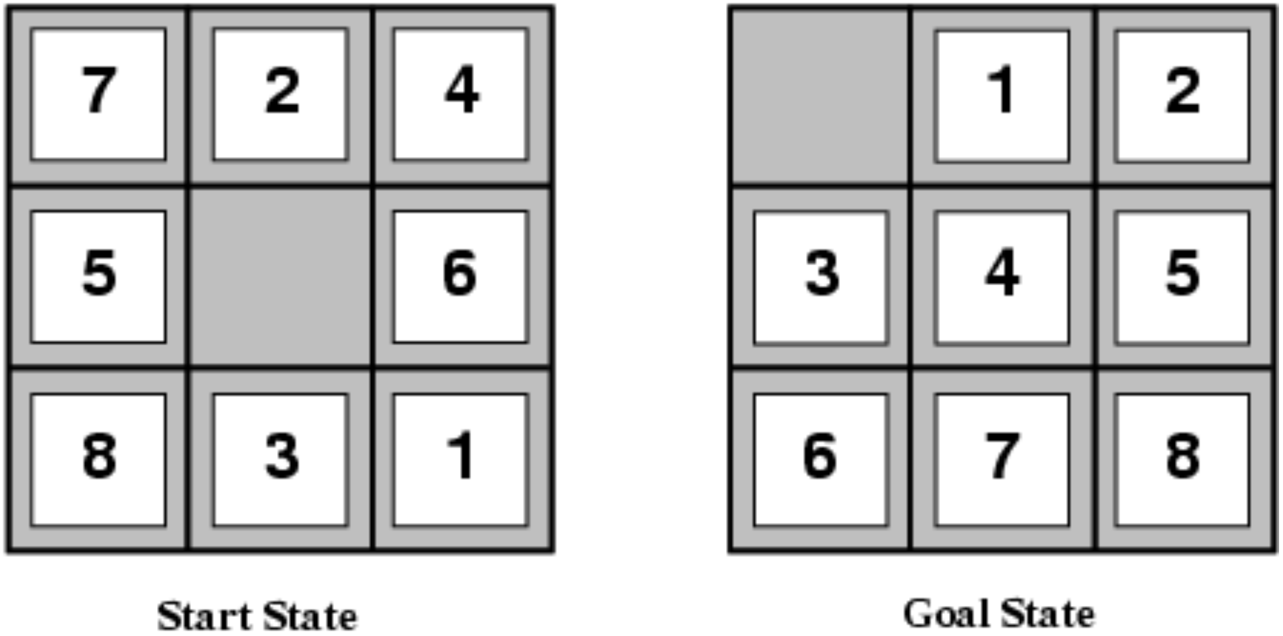
- Unfortunately
  - space is a problem—keeps all nodes in memory
  - Time is typically exponential since number of states in “goal contour” is usually exponential in the length of the solution

- Book has a couple A* variations that attempt to restrict space usage

Admissible heuristics for 8-puzzle

- $h_1(n)$ = number of misplaced tiles
- $h_2(n)$ = total Manhattan distance

(i.e., no. of squares from desired location of each tile)



- $h_1(S) = ?$ 8
- $h_2(S) = ?$ 3+1+2+2+2+3+3+2 = 18

Dominance

- If $h_2(n) \geq h_1(n)$ for all n (both admissible)
- then h_2 **dominates** h_1
- h_2 is better for search

- Typical search costs (average number of nodes expanded):

- *depth=12*

  IDS generates 3644035 nodes
  $A^*(h_1)$ generates 227 nodes
  $A^*(h_2)$ generates 73 nodes

- *depth=24*

  IDS generates too many nodes!
  $A^*(h_1)$ generates 39135 nodes
  $A^*(h_2)$ generates 1641 nodes