Lecture 3

CS 6260 Automated Planning and Learning

Informed Search 2: A* Search

Department of Computer Science and Engineering

Indian Institute of Technology Madras

Some material based on lecture slides for ASU's *CSE571 Artificial Intelligence by Siddharth Srivastava*.

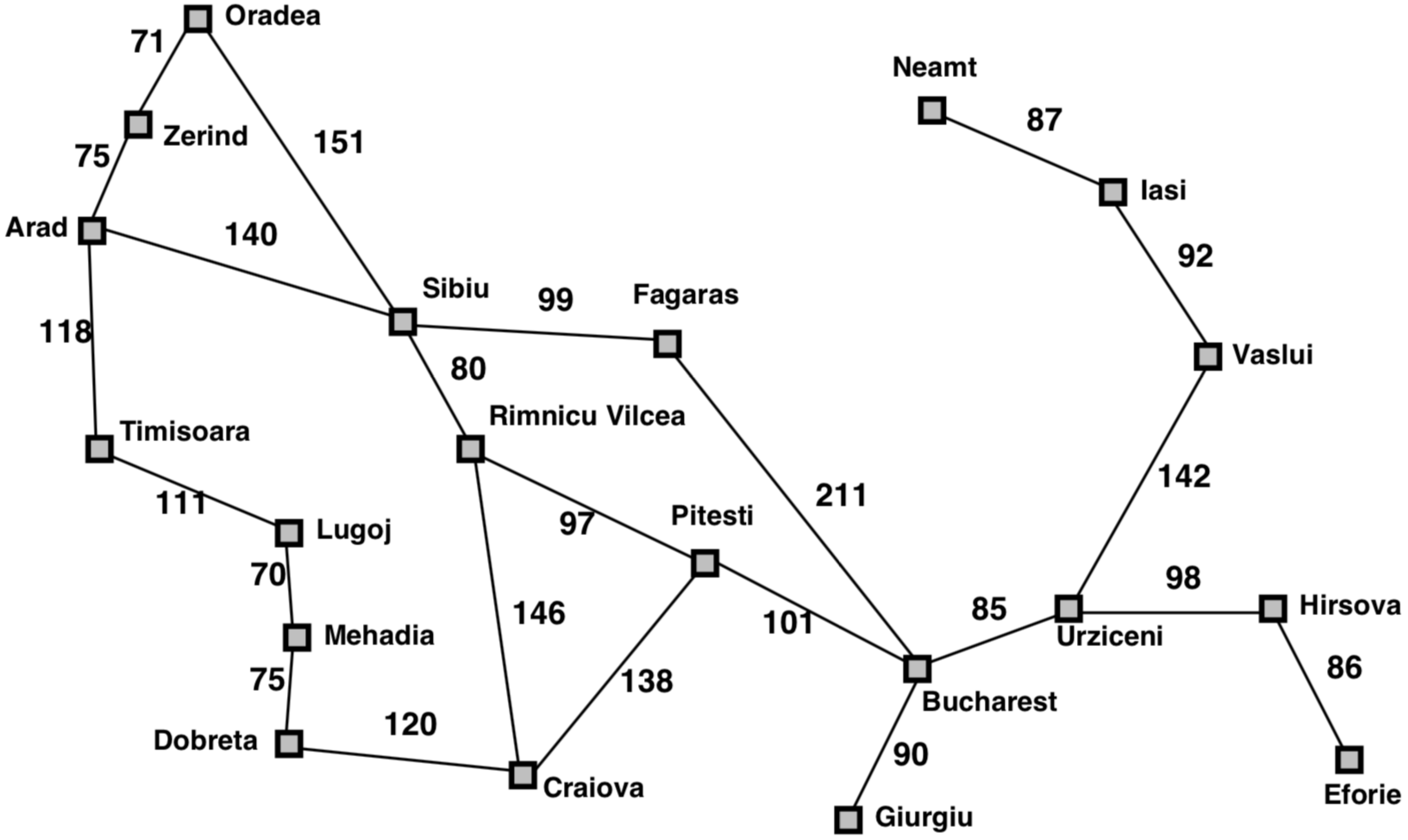
Recap: Informed Search

Informed Search

- Can we use goal-cost estimates?
  - This cost estimate is known as a heuristic function
  - Cost estimates amount to measuring the desirability of a state

- Informed search algorithms use the heuristic to guide node expansion
  - Fringe is implemented as a priority queue
  - Key for the queue is computed *using the heuristic function*

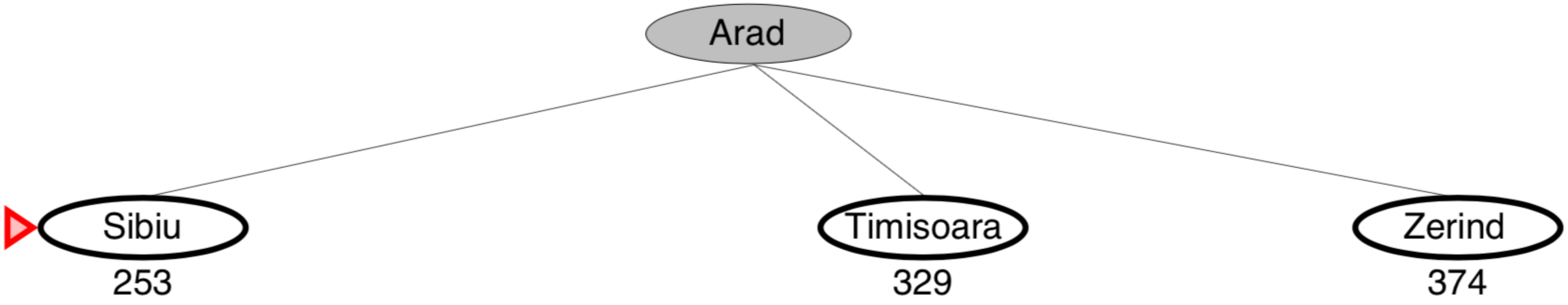
Heuristic: Example

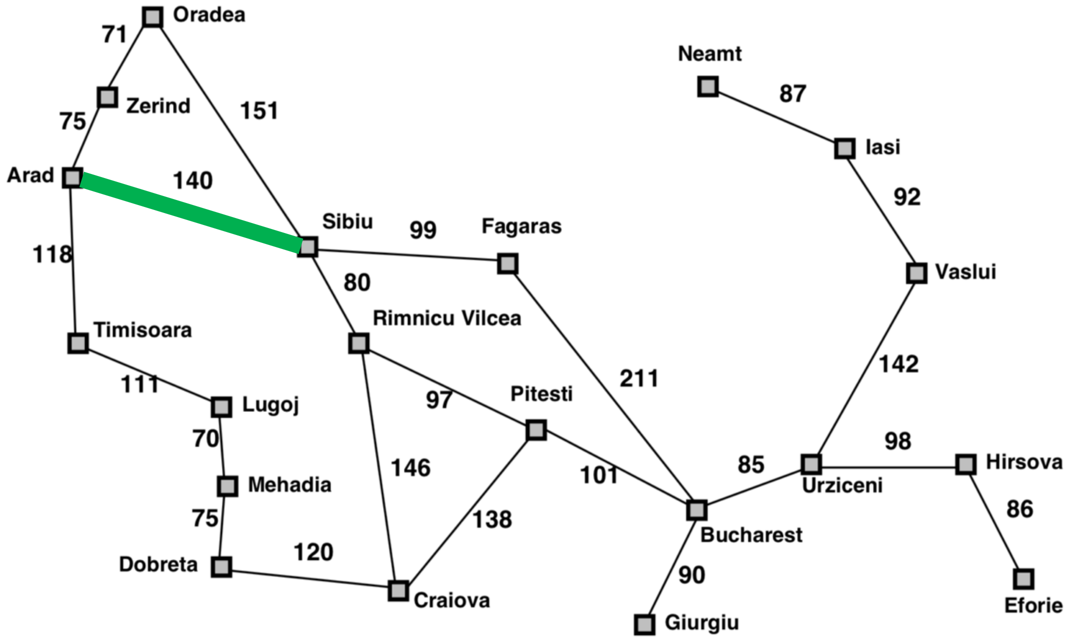


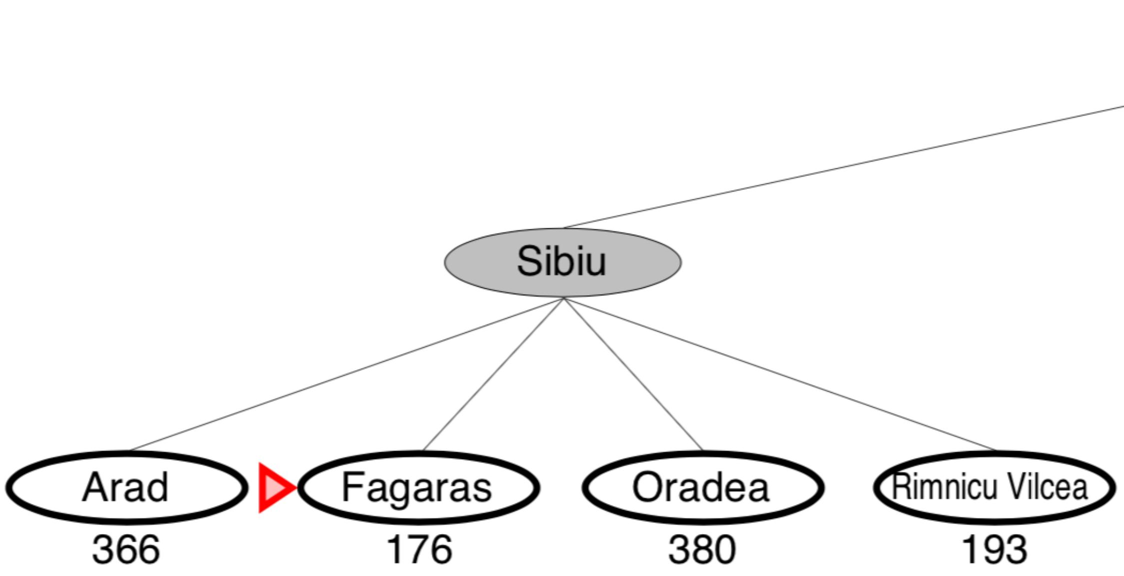
Straight−line distance
to Bucharest

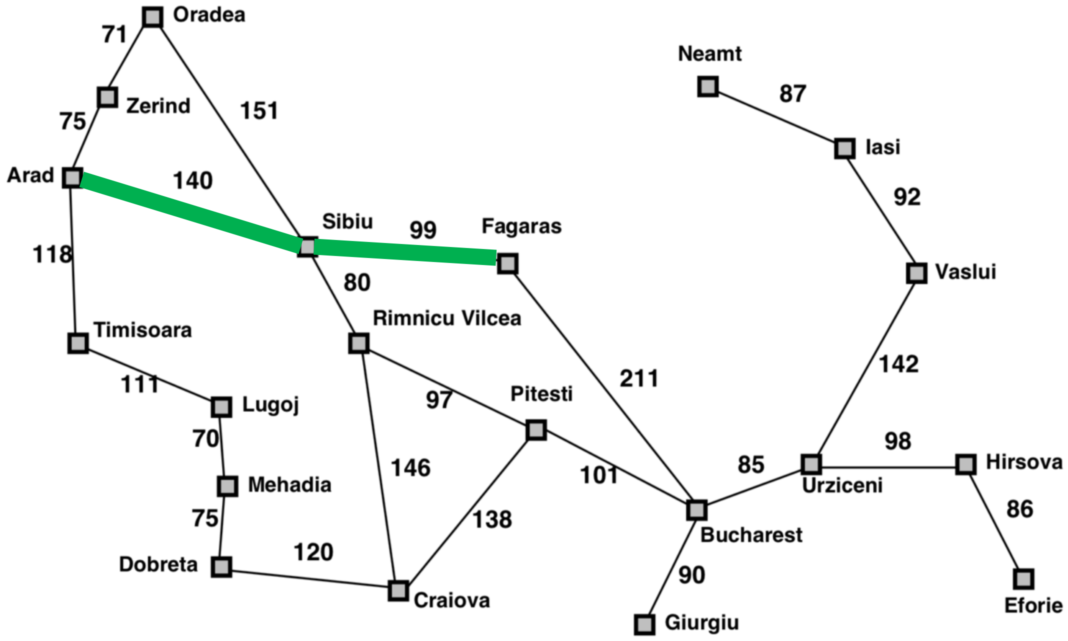
Arad	366
Bucharest	0
Craiova	160
Dobreta	242
Eforie	161
Fagaras	178
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadia	241
Neamt	234
Oradea	380
Pitesti	98
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

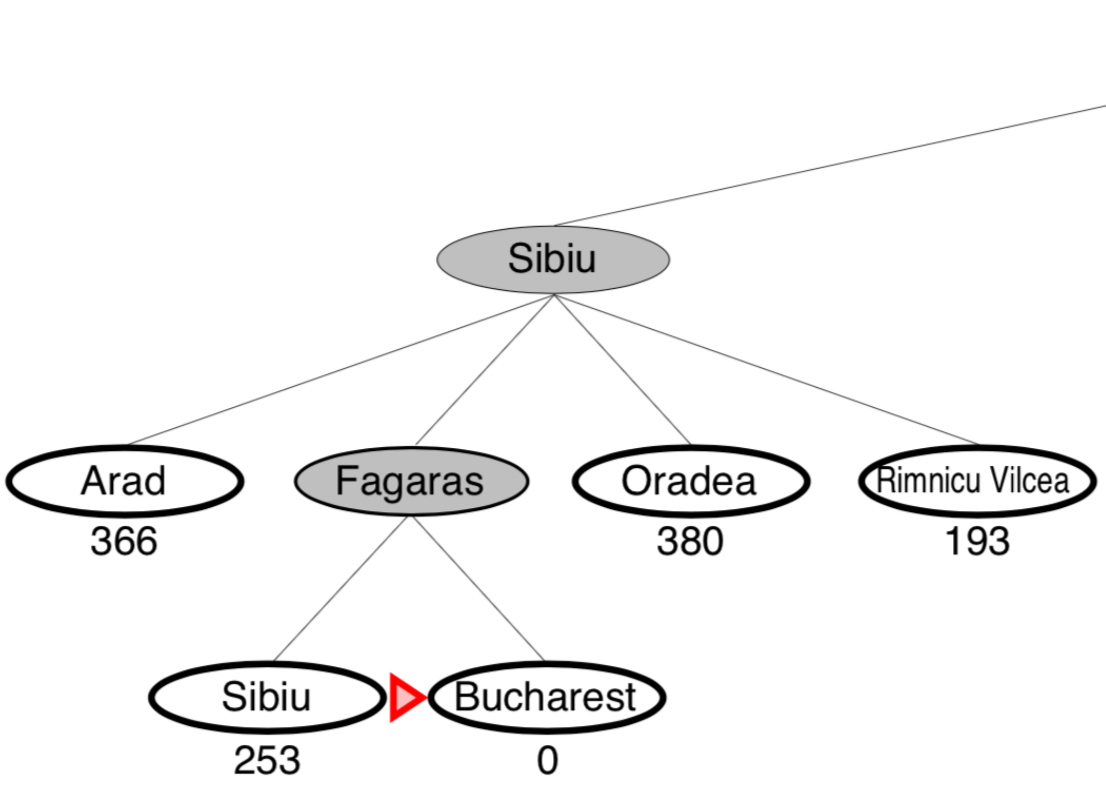

Arad
366

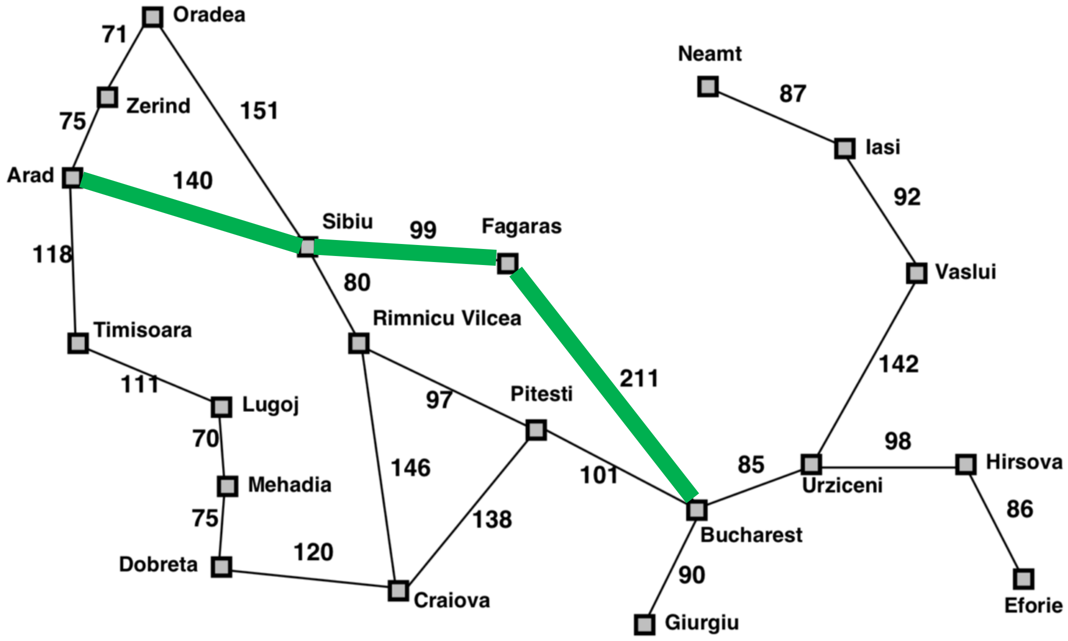

Arad
Sibiu
253
Timisoara
329
Zerind
374


Oradea
71
Neamt
Zerind
87
151
75
Iasi
Arad
140
92
Sibiu
99
Fagaras
118
Vaslui
80
Rimnicu Vilcea
Timisoara
142
211
111
Lugoj
97
Pitesti
70
98
Mehadia
146
101
85
Urziceni
Hirsova
86
75
138
Bucharest
Dobreta
120
90
Craiova
Eforie
Giurgiu


Arad
Sibiu
Timisoara
329
Zerind
374
Arad
366
Fagaras
176
Oradea
380
Rimnicu Vilcea
193


Oradea
71
Neamt
Zerind
87
75
151
Iasi
Arad
140
92
Sibiu
99
Fagaras
118
80
Vaslui
Rimnicu Vilcea
Timisoara
142
111
211
Lugoj
97
Pitesti
70
98
Mehadia
146
85
Hirsova
101
Urziceni
75
138
86
Bucharest
Dobreta
120
90
Craiova
Eforie
Giurgiu


Arad
Sibiu
Timisoara
329
Zerind
374
Arad
366
Fagaras
Oradea
380
Rimnicu Vilcea
193
Sibiu
253
Bucharest
0


Oradea
71
Neamt
Zerind
87
75
151
Iasi
Arad
140
92
Sibiu
99
Fagaras
118
80
Vaslui
Rimnicu Vilcea
Timisoara
142
111
211
Lugoj
97
Pitesti
70
98
Mehadia
146
101
85
Urziceni
Hirsova
86
75
138
Bucharest
Dobreta
120
90
Craiova
Eforie
Giurgiu

Greedy Best First Search Properties

- Complete in finite graphs with closed list
- Time
- Space
- Optimal?

b = Max Breadth
m = Max Depth

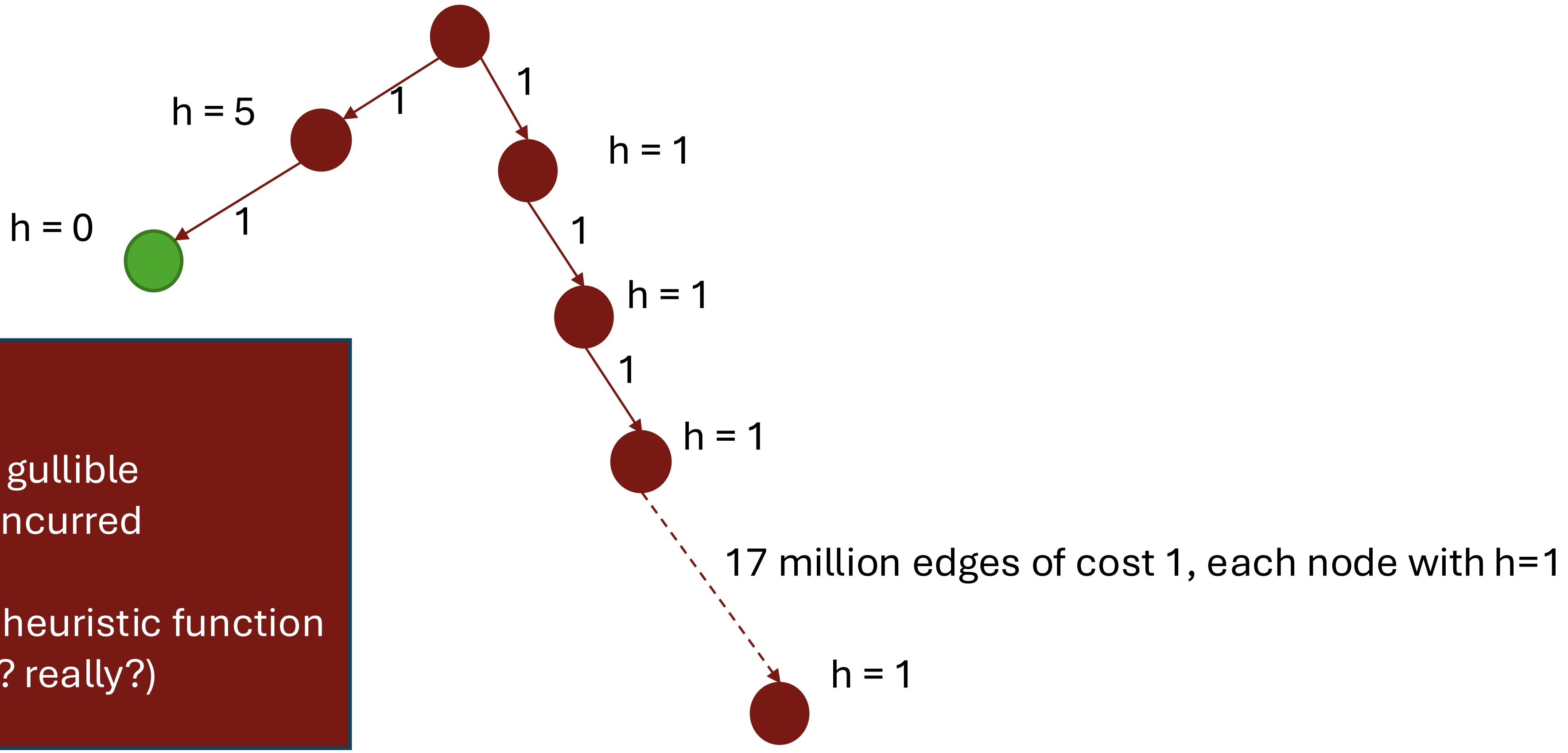
Greedy Best First Search Properties

- Complete in finite graphs with closed list
- Time $O(b^m)$
- Space $O(b^m)$
- Optimal? No

What if the heuristic is not very good?

Greedy Search: Negative Result

- In the worst case, greedy search can expand all other nodes in the graph (even those that are arbitrarily further than the goal) before reaching the goal



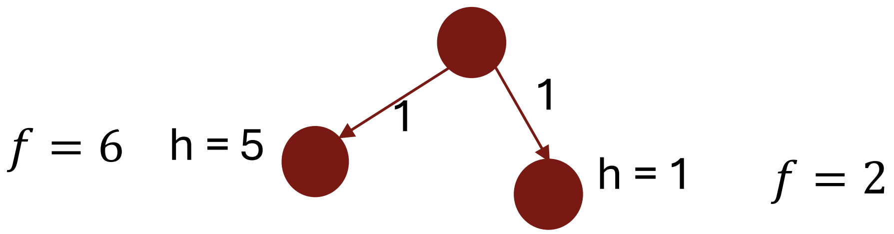
Oops!

- Algorithm could be made less gullible
  - Pay attention to the cost incurred
    ~ gambler's ruin
- Our stated requirements for a heuristic function were too loose (anything goes? really?)

A* Search

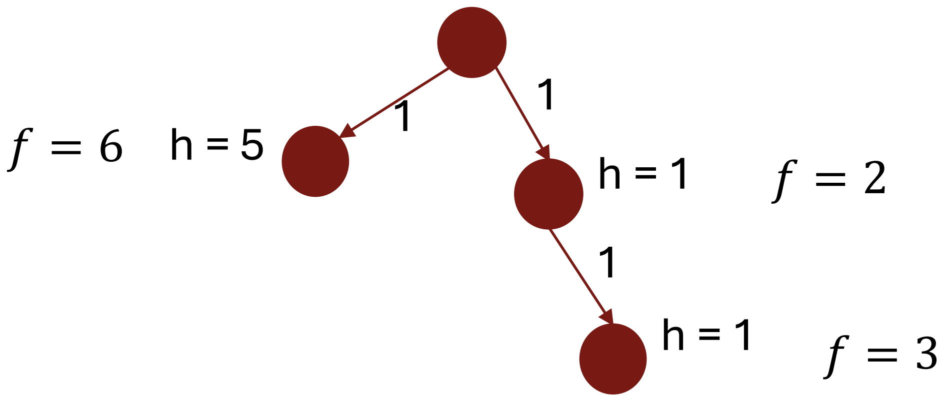
A* Search

- A* directly addresses those limitations
- $g(n)$ = cost required to get to the node from the start
- $h(n)$ = estimated cost to the goal from n (heuristic, as before)
- $f(n) = g(n) + h(n)$
- Expansion strategy:
  - expand the node with the least f value



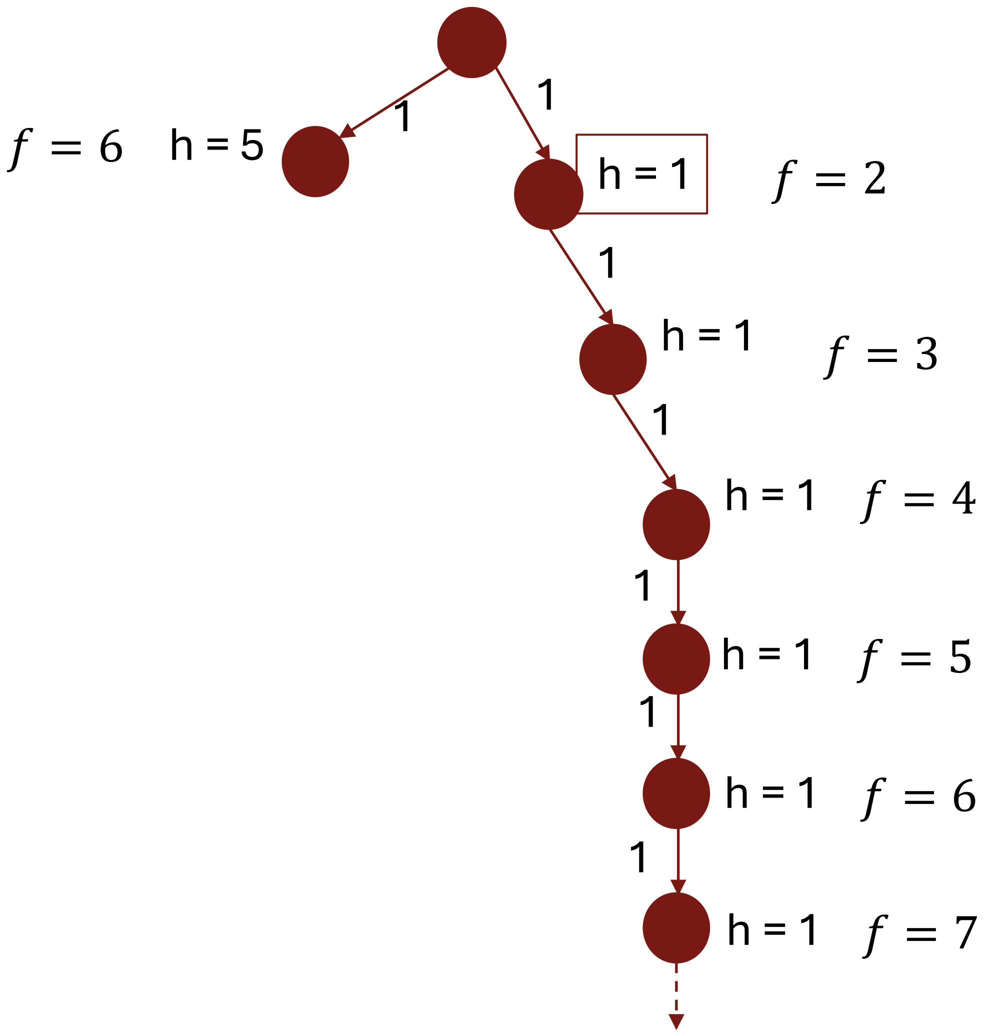
A* Search

- A* directly addresses those limitations
- $g(n)$ = cost required to get to the node from the start
- $h(n)$ = estimated cost to the goal from n (heuristic, as before)
- $f(n) = g(n) + h(n)$
- Expansion strategy:
  - expand the node with the least f value



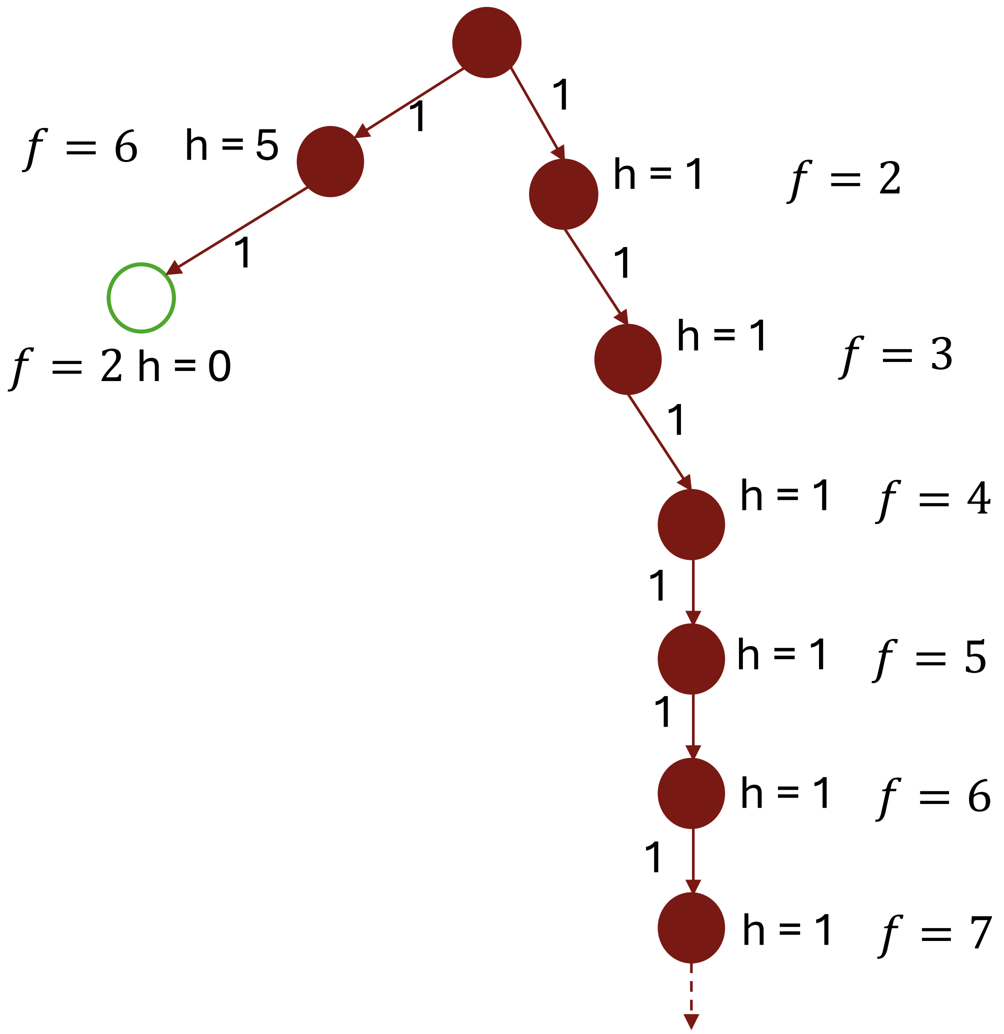
A* Search

- A* directly addresses those limitations
- $g(n)$ = cost required to get to the node from the start
- $h(n)$ = estimated cost to the goal from n (heuristic, as before)
- $f(n) = g(n) + h(n)$
- Expansion strategy:
  - expand the node with the least f value



A* Search

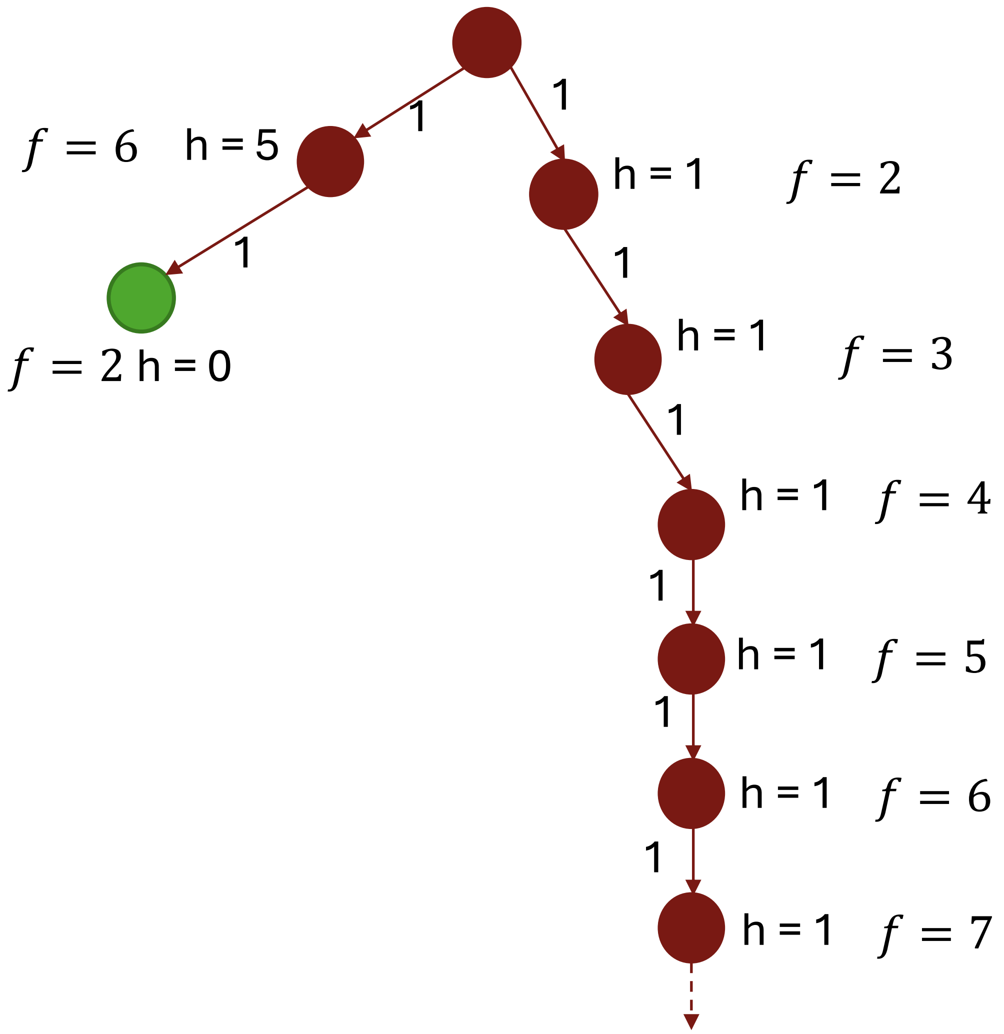
- A* directly addresses those limitations
- $g(n)$ = cost required to get to the node from the start
- $h(n)$ = estimated cost to the goal from n (heuristic, as before)
- $f(n) = g(n) + h(n)$
- Expansion strategy:
  - expand the node with the least f value



A* Search

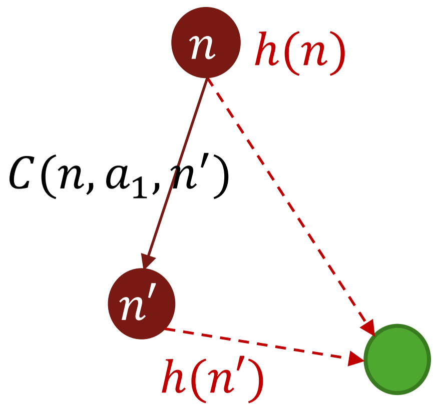
- A* directly addresses those limitations
- $g(n)$ = cost required to get to the node from the start
- $h(n)$ = estimated cost to the goal from n (heuristic, as before)
- $f(n) = g(n) + h(n)$
- Expansion strategy:
  - expand the node with the least f value

Not immune to bad h
But won't get fooled forever



Optimality of A*

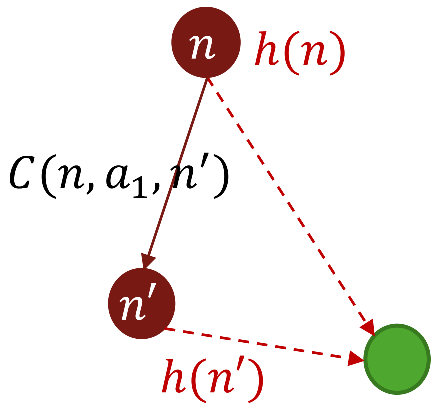
- A heuristic is admissible if
  - $h(n) \leq h^*(n)$, where $h^*(n)$ is the optimal cost of reaching the goal from n
  - $h(g) = 0$
  - Admissible heuristics are optimistic about the cost to the goal
- A heuristic is consistent if for every a
  - $h(n) - h(n') \leq C(n, a, n')$
  - Consistent heuristics are optimistic about the cost of each edge
  - This is a more specific constraint



What is the relationship between admissible and consistent heuristics?

Optimality of A*

- A heuristic is admissible if
  - $h(n) \leq h^*(n)$, where $h^*(n)$ is the optimal cost of reaching the goal from n
  - $h(g) = 0$
  - Admissible heuristics are optimistic about the cost to the goal
- A heuristic is consistent if for every a
  - $h(n) - h(n') \leq C(n, a, n')$
  - Consistent heuristics are optimistic about the cost of each edge
  - This is a more specific constraint



- When using a consistent heuristic, graph search and tree search versions of A* are optimal
- When using an admissible but inconsistent heuristic,
  - tree search version of A* is optimal
  - graph search version is optimal when closed list is managed carefully

Proof Exercise

- To Prove: Let h be a consistent heuristic that is 0 for goal states. Then h is admissible.
- Proof:

  Let h be a consistent heuristic.

  This implies $h(n) - h(n') \leq C(n, a, n')$, for all a (by definition)... (1)

  Let π be a path from $s_0 \xrightarrow{a_1} s_1 \xrightarrow{a_2} s_2 \cdots \xrightarrow{a_m} s_m \xrightarrow{a_g} s_g$, where s_g is a goal node

  $C(\pi) = C(s_o, a_1, s_1) + C(s_1, a_2, s_2) + \cdots + C(s_{m-1}, a_m, s_m) + C(s_m, a_g, s_g)$ (by definition of path cost)

  $\geq h(s_0) - h(s_1) + h(s_1) - h(s_2) + \cdots + h(s_{m-1}) - h(s_m) + h(s_m) - h(s_g)$ (using 1)

  $= h(s_0) - h(s_g)$

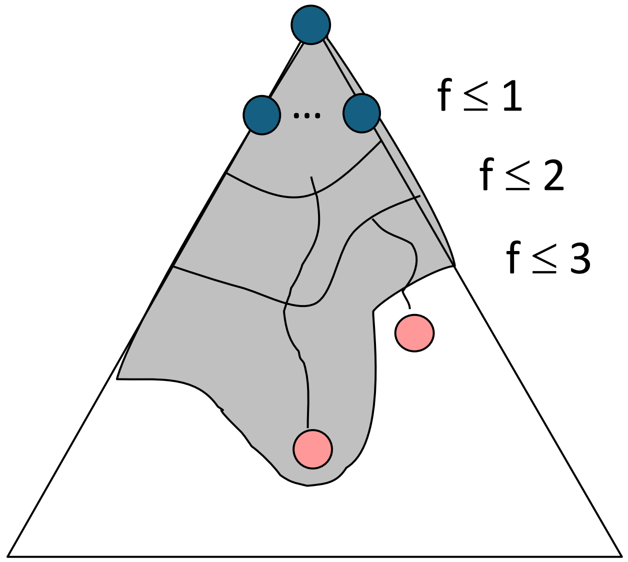
  $= h(s_0)$ (because $h(s_g) = 0$, given)

  Thus, $h(s_0) \leq C(\pi)$ for any path π from s_0 to goal and in particular for $\pi = \pi^*$, the optimal path from s_0 to a goal

  Thus, h is admissible (by definition of admissibility)

Why is A* Optimal for Consistent Heuristics?

- Lemma 1: If h is consistent and then the f values along any path generated by A* are non-decreasing
  - (proof similar to preceding proof)
- Lemma 2: A* expands nodes in non-decreasing order of their f values
  - By definition of the expansion strategy
- Thus, if A* expands a goal node, by Lemma 2, there cannot be a future expansion that has a lower f value
  - When a goal node is expanded, $f =$ distance to that goal *node*
  - *No future expansion of the goal node will have a lower value of f*
- Q: Will it necessarily expand a goal node if one is reachable?
  - If there are only finitely many nodes n such that $f(n) < C^*$



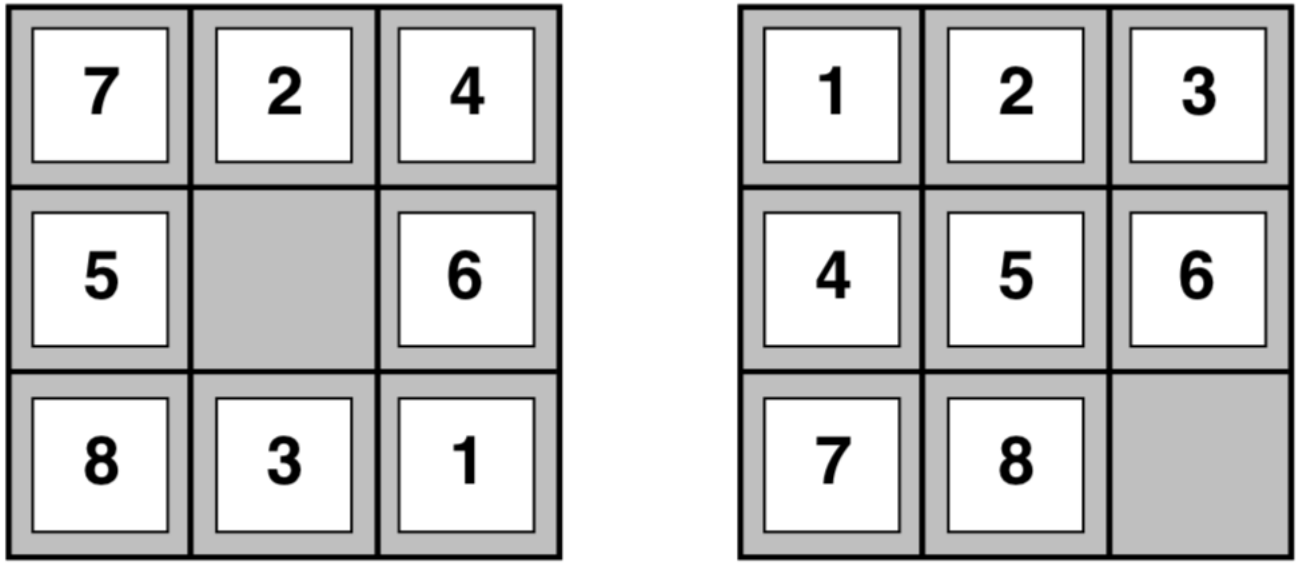
A* Properties

- Completeness: Yes, unless there are infinitely many nodes with $f \leq C^*$
- Time Complexity: Exponential in relative error $\times$ length of solution
- Space Complexity: $O(b^m)$
- Optimal: Yes (conditions on h)

Generating Heuristics

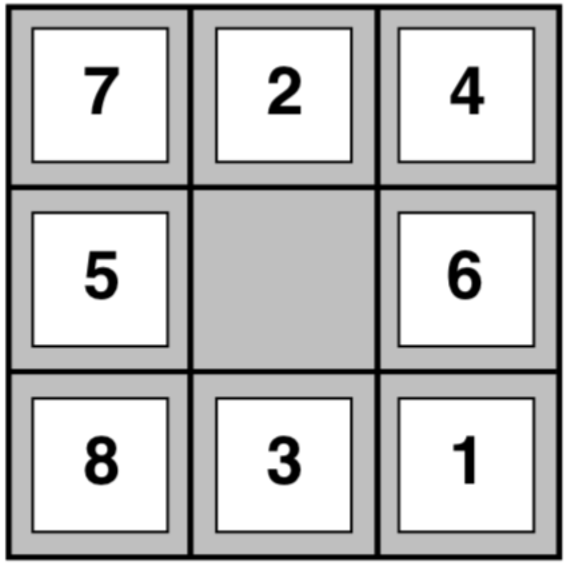
Heuristic Estimates

- Easier in some problems than others!
  - Route finding problem has a natural metric (straight line distance)
  - Most state spaces are not so fortunate
- Online shopping agent
  - State space…
  - Metric?
- Household robot that enters a dark room
  - ”Distance” between a dark room and the same room with the light on

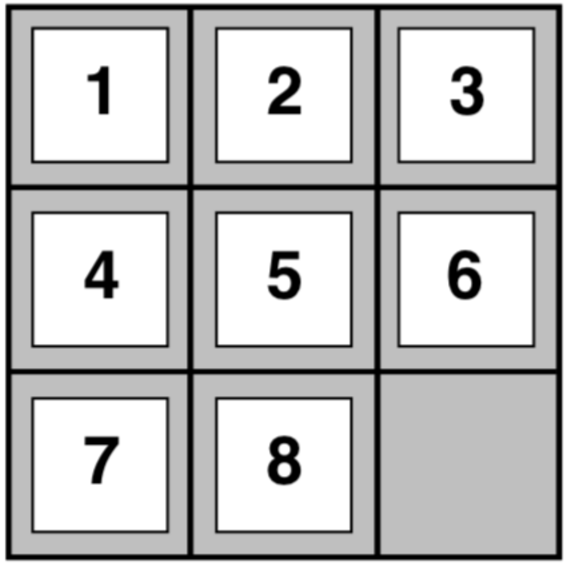


Start State Goal State

- Can you think of an admissible heuristic?



Start State



Goal State

- Can you think of an admissible heuristic?
  - Number of misplaced tiles (6)
  - Total Manhattan distance $(4 + 0 + 3 + 3 + 1 + 0 + 2 + 1 = 14)$

Methodology for Generating Heuristics

- Create a relaxed version of the problem
  - Abstract, or approximate and easier
- Relaxed problems allow shortcuts
  - Optimal solution cost in relaxed $\leq$ optimal solution cost in real
- Optimal solution cost in the relaxed problem constitutes a heuristic
- But, if the relaxed problem is not easy, computing its optimal can be a challenge!

Where A* is Used in Practice…

- Video game route finding problems
- Robot path planning
- Natural Language Processing
- Speech recognition

Search Algorithms: Summary

- Use successor functions to formulate large search problems
- General notion of tree search and graph search
  - Fringe,
  - Node expansion strategies
- Informed vs uninformed search
  - If you can get a cost-to-goal estimate, use it as a heuristic
  - Admissible heuristics are good for guaranteed optimality
- Read further to learn about using inadmissible heuristics
  - (What would A* do?)
- To see some of these algorithms in action: https://movingai.com/SAS/