Announcements

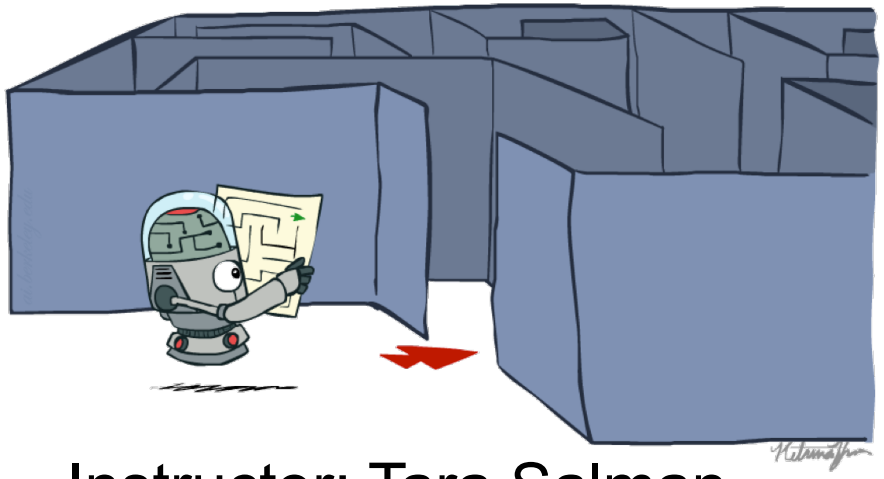
- Homework 0 and Project 0
  - Project 0 submission via blackboard
  - Homework 0 via gradescope
  - Due **Tuesday, Sep 7th, at 9:30am**

- Homework 0 is optional
  - To motivate you to solve it, you can take up to 2 additional grades for homework if you solve it.

- Common mistakes
  - HW0: Not following the guidelines on how to write answers
    - E.g., 0.0001 instead of .0001

CS 3568: Intelligent Systems

Search (Part 3)



Instructor: Tara Salman

Texas Tech University

Computer Science Department

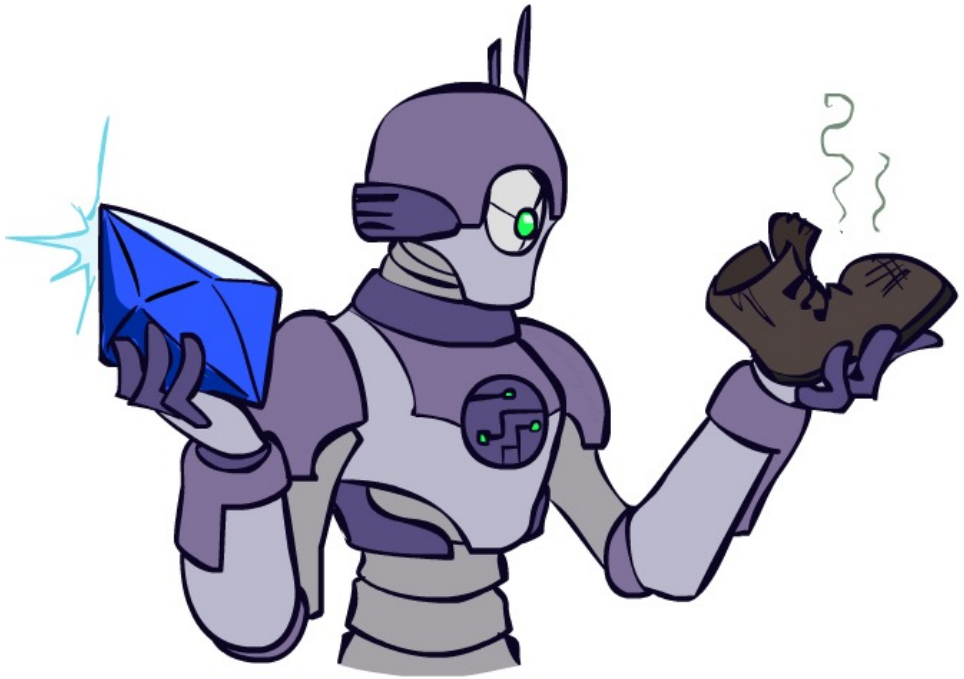
[These slides were created by Dan Klein and Pieter Abbeel for CS188 Intro to AI at UC Berkeley (ai.berkeley.edu).]

Last Lecture

- Uniformed Search
  - BFS
  - Cost Sensitive search
- Informed Search
  - Heuristics
  - Greedy Search

Today

- Informed Search
  - A* Search
- Optimality of A*
- How to choose heuristics
- Graph Search



A* Search

A* Search

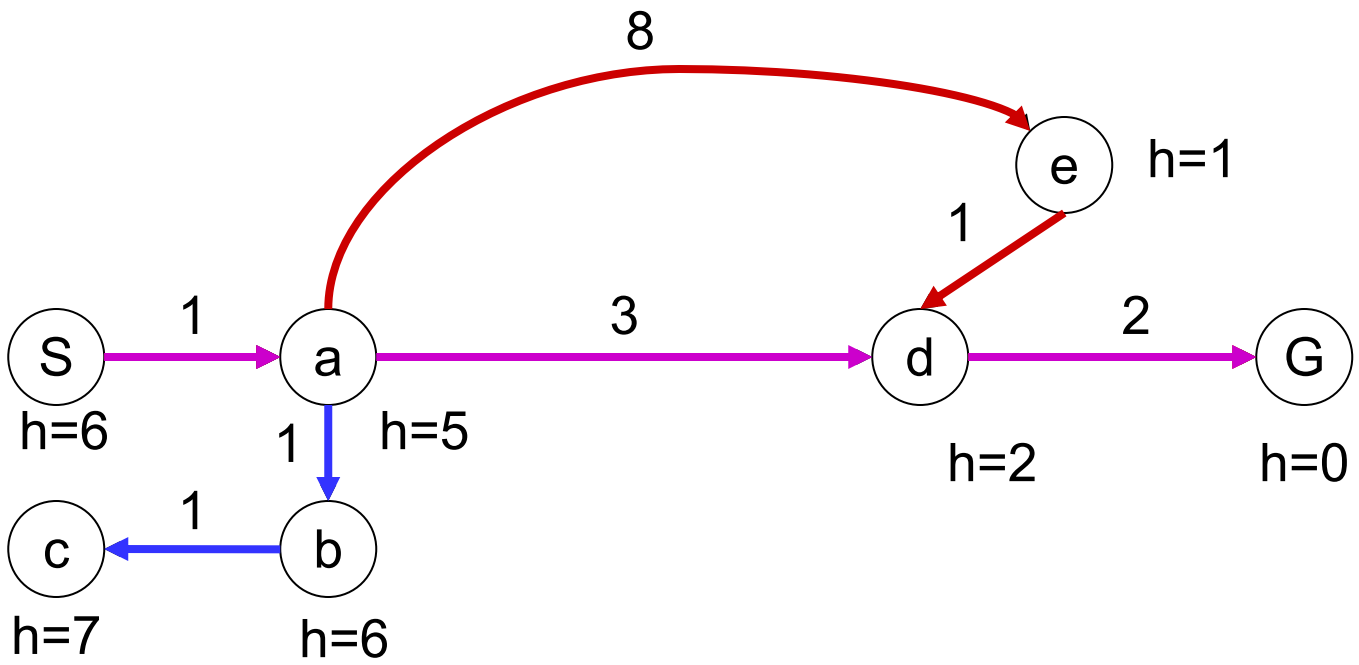
UCS

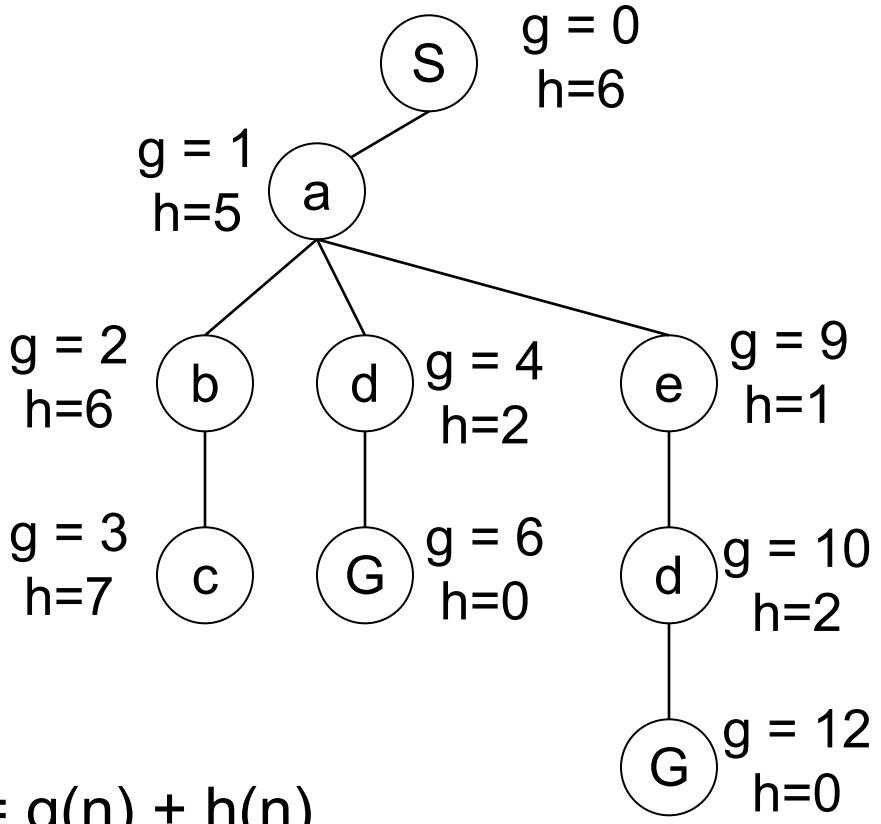
Greedy

A*

Combining UCS and Greedy

- Uniform-cost orders by path cost, or backward cost g(n)
- Greedy orders by goal proximity, or forward cost h(n)

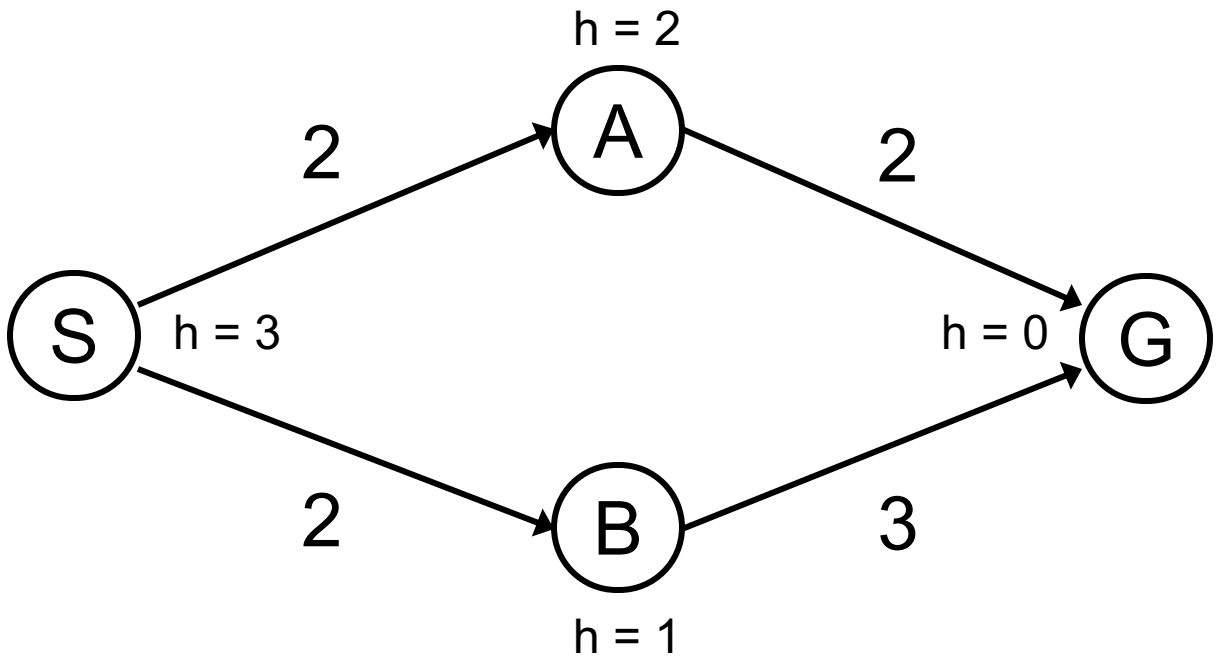




- A* Search orders by the sum: f(n) = g(n) + h(n)

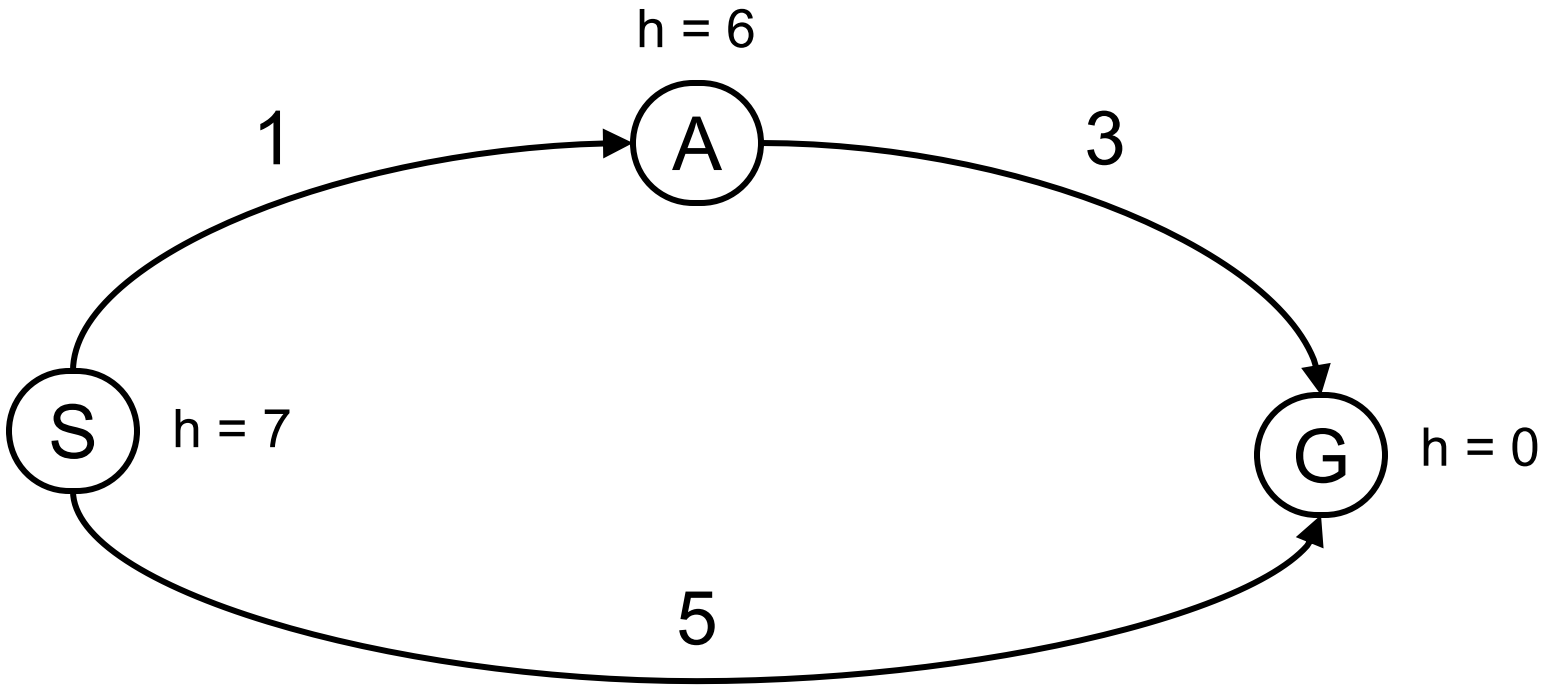
When should A* terminate?

- Should we stop when we enqueue a goal?



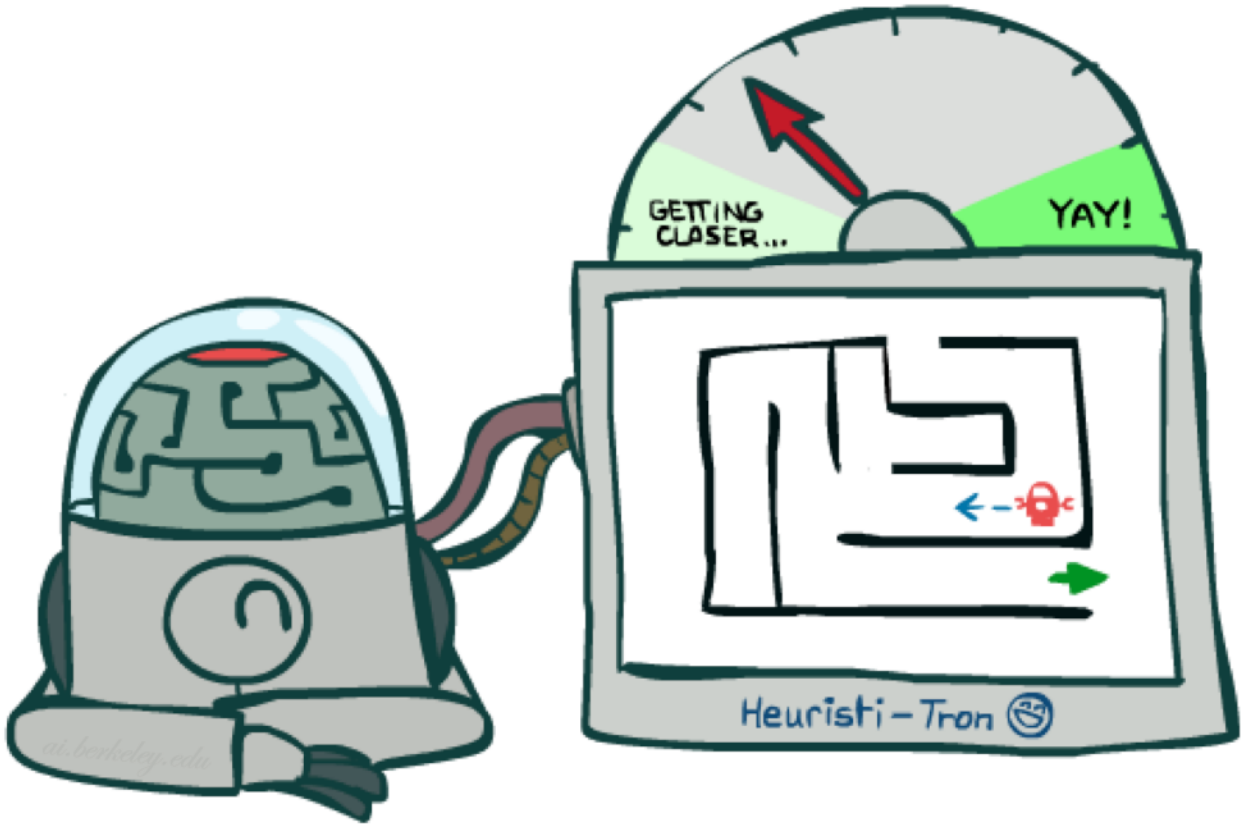
- No: only stop when we dequeue a goal

Is A* Optimal?

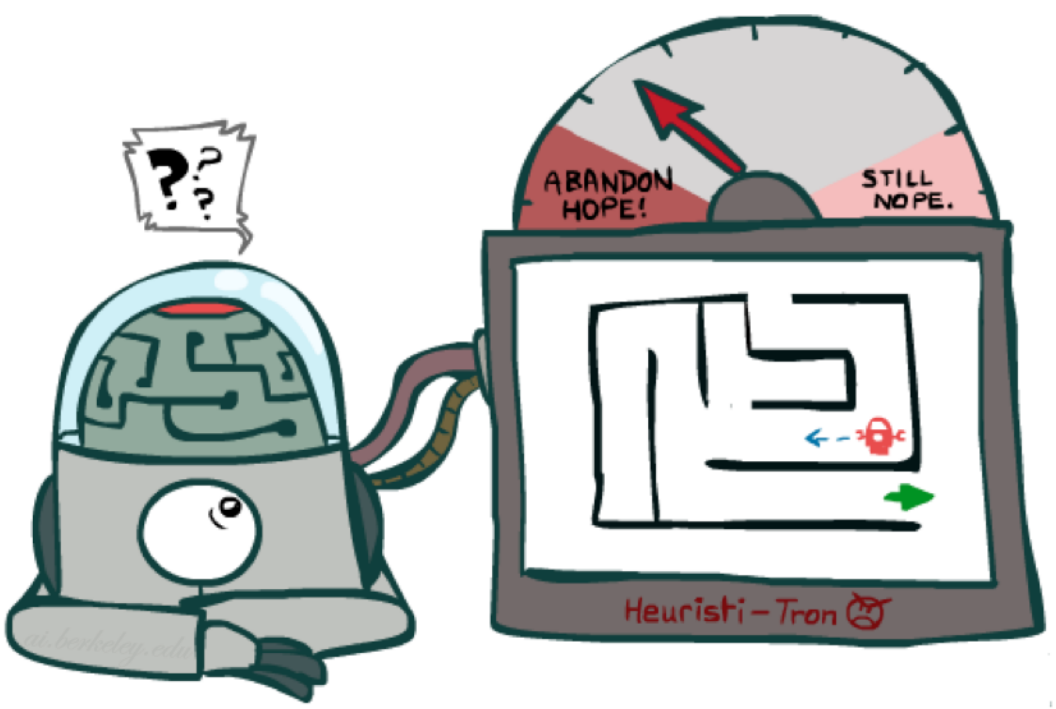


- What went wrong?
- Actual bad goal cost < estimated good goal cost (from heuristic)
- We need estimates to be less than actual costs!

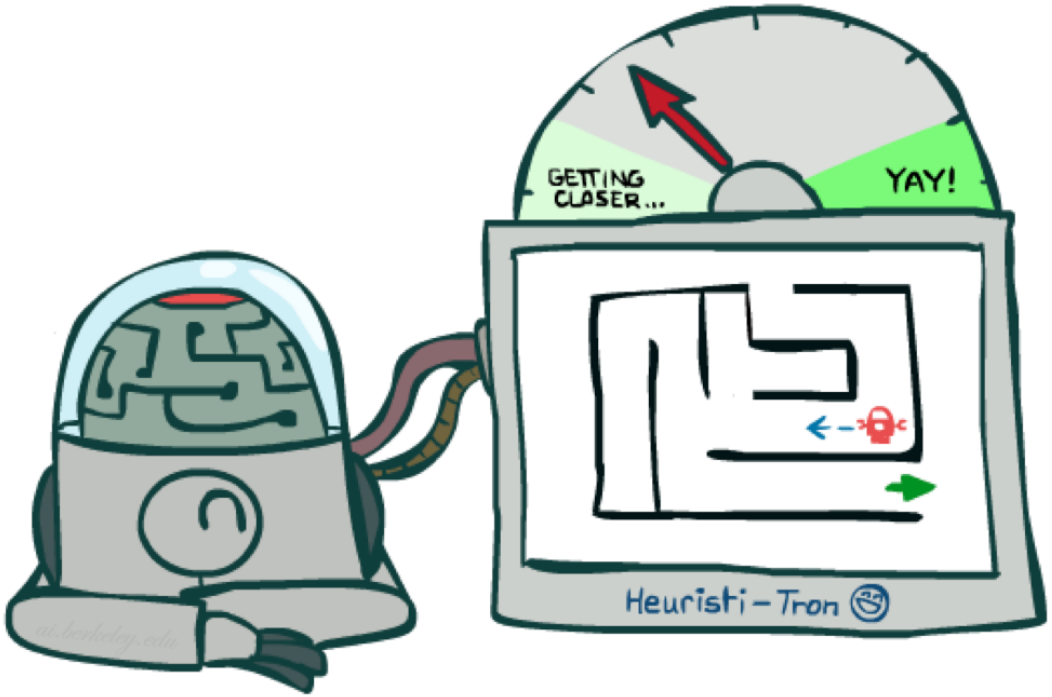
Admissible Heuristics



Idea: Admissibility



Inadmissible (pessimistic) heuristics break optimality by trapping good plans on the fringe



Admissible (optimistic) heuristics slow down bad plans but never outweigh true costs

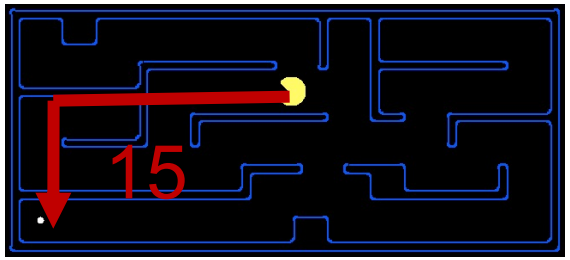
Admissible Heuristics

- A heuristic h is *admissible* (optimistic) if:

$$0 \leq h(n) \leq h^*(n)$$

where $h^*(n)$ is the true cost to a nearest goal

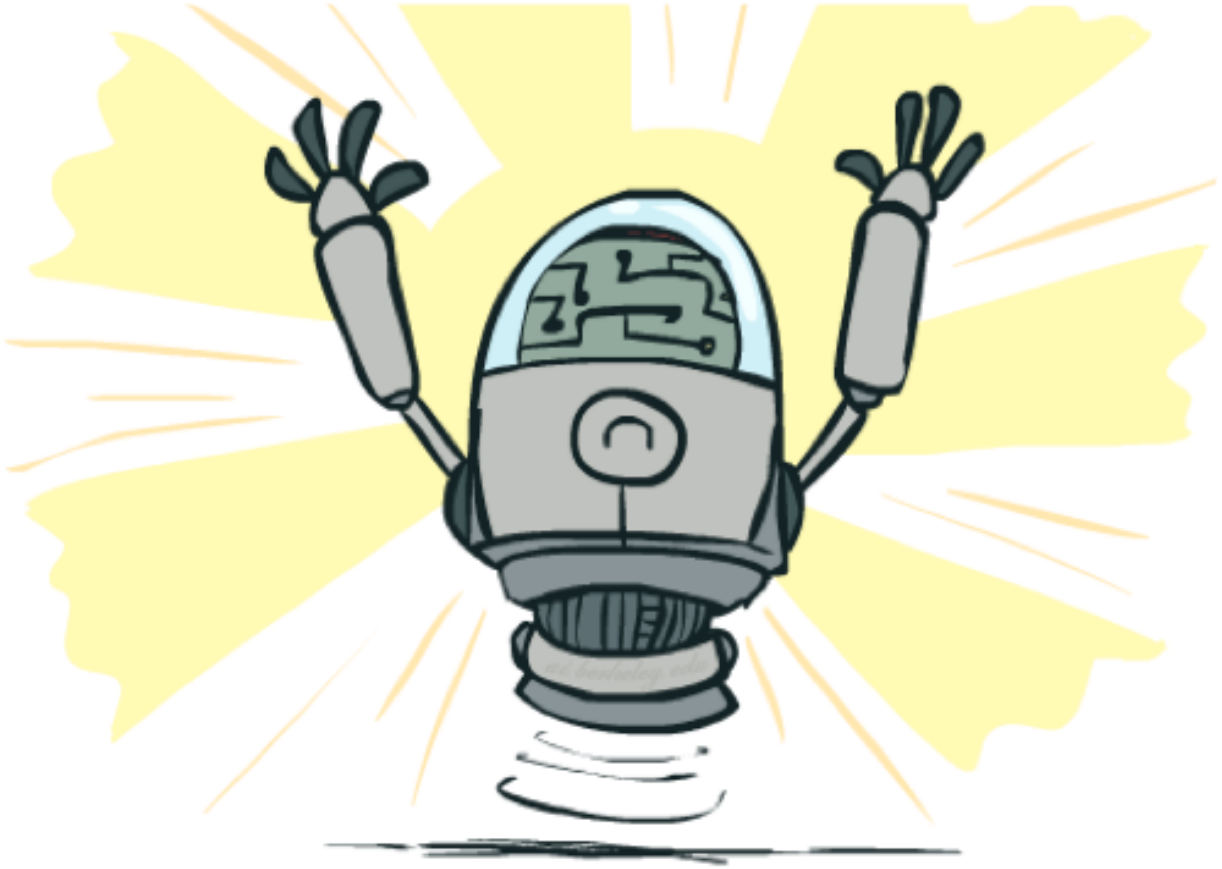
- Examples:



4

- Coming up with admissible heuristics is most of what's involved in using A* in practice.

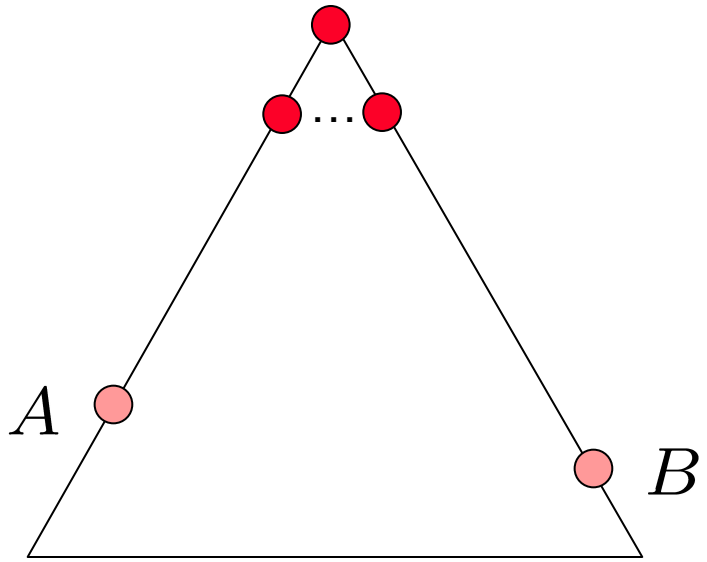
Optimality of A* Tree Search



Optimality of A* Tree Search

Assume:

- A is an optimal goal node
- B is a suboptimal goal node
- h is admissible



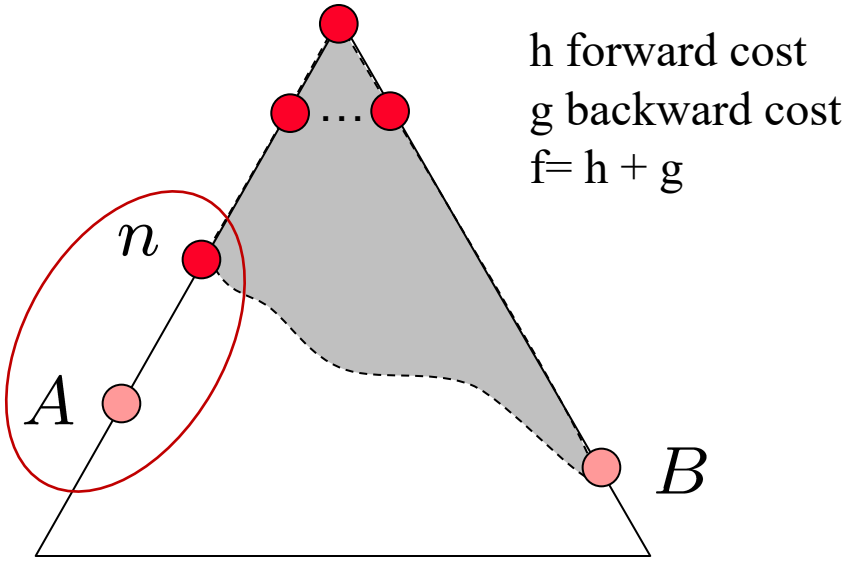
Claim:

- A will exit the fringe before B

Optimality of A* Tree Search: Blocking

Proof:

- Imagine B is on the fringe
- Some ancestor n of A is on the fringe, too (maybe A!)
- Claim: n will be expanded before B
  1. f(n) is less or equal to f(A)



$$f(n) = g(n) + h(n)$$ Definition of f-cost

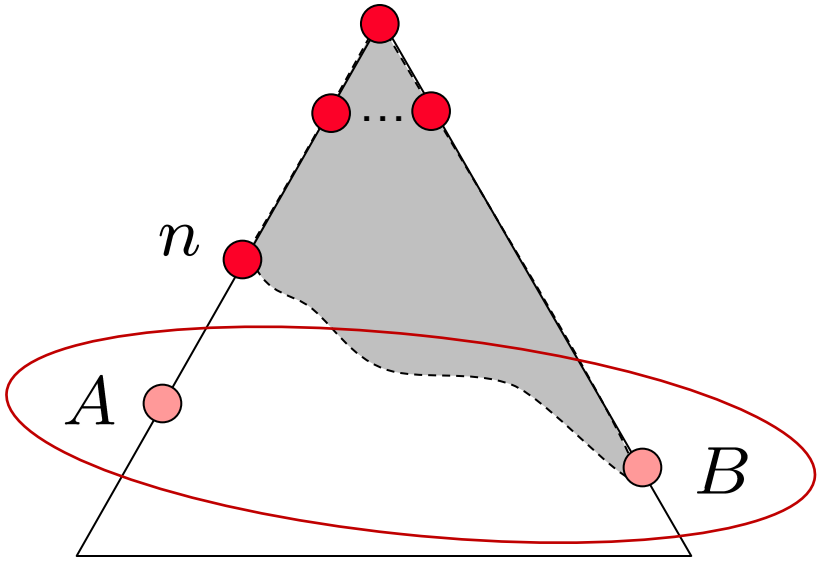
$$f(n) \leq g(A)$$ Admissibility of h

$$g(A) = f(A)$$ h = 0 at a goal

Optimality of A* Tree Search: Blocking

Proof:

- Imagine B is on the fringe
- Some ancestor n of A is on the fringe, too (maybe A!)
- Claim: n will be expanded before B
  1. f(n) is less or equal to f(A)
  2. f(A) is less than f(B)



$$g(A) < g(B)$$
$$f(A) < f(B)$$

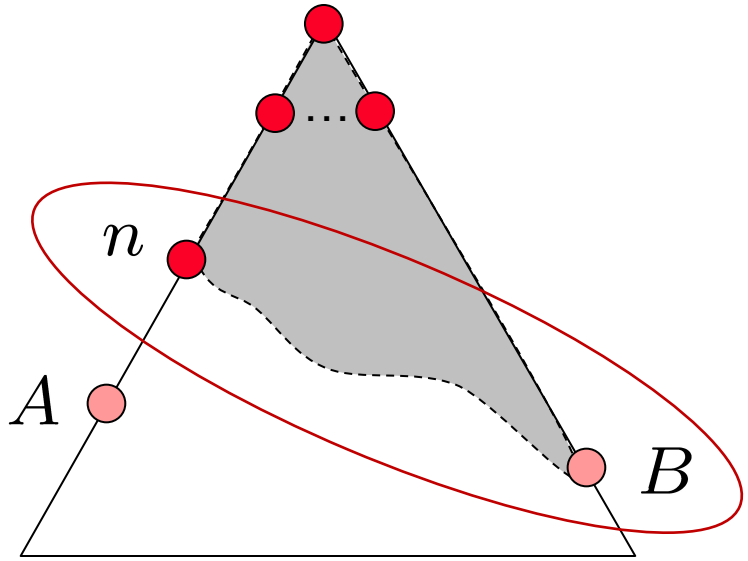
B is suboptimal

h = 0 at a goal

Optimality of A* Tree Search: Blocking

Proof:

- Imagine B is on the fringe
- Some ancestor n of A is on the fringe, too (maybe A!)
- Claim: n will be expanded before B
  1. f(n) is less or equal to f(A)
  2. f(A) is less than f(B)
  3. n expands before B
- All ancestors of A expand before B
- A expands before B
- A* search is optimal

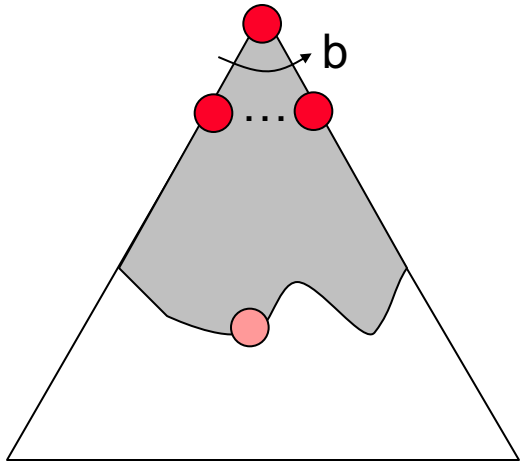


$$f(n) \leq f(A) < f(B)$$

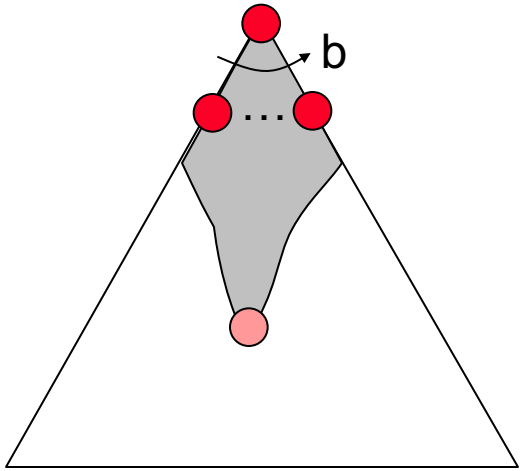
Properties of A*

Properties of A*

Uniform-Cost

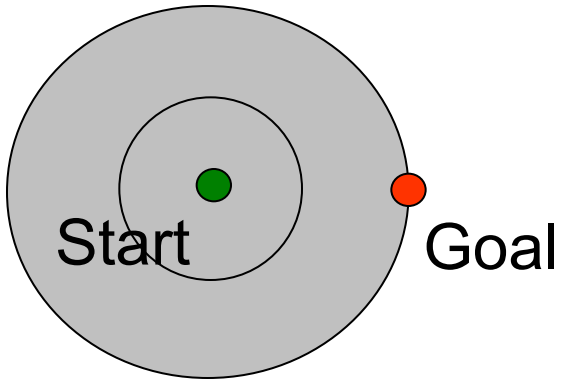


A*

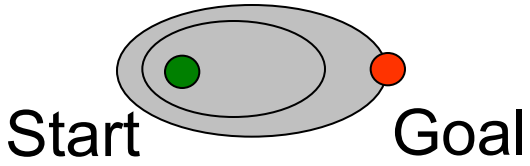


UCS vs A* Contours

- Uniform-cost expands equally in all "directions"



- A* expands mainly toward the goal, but does hedge its bets to ensure optimality



Comparison

Greedy

Uniform Cost

A*

A* Applications

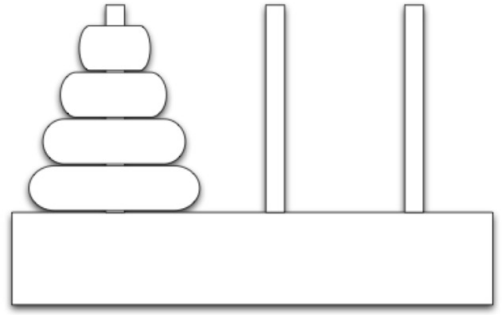
A* Applications

- Video games
- Pathing / routing problems
- Resource planning problems
- Robot motion planning
- Language analysis
- Machine translation
- Speech recognition
- …

Review Slides for all search problems

Review (Search Problem formulation)

- The Towers of Hanoi is a famous problem for studying recursion in computer science and recurrence equations in discrete mathematics. We start with N discs of varying sizes on a peg (stacked in order according to size),and two empty pegs. We are allowed to move a disc from one peg to another, but we are never allowed to move a larger disc on top of a smaller disc. The goal is to move all the discs to the rightmost peg (see figure).
  - Propose a state representation for the problem
  - What is the start state?
  - From a given state, what actions are legal?
  - What is the goal test?



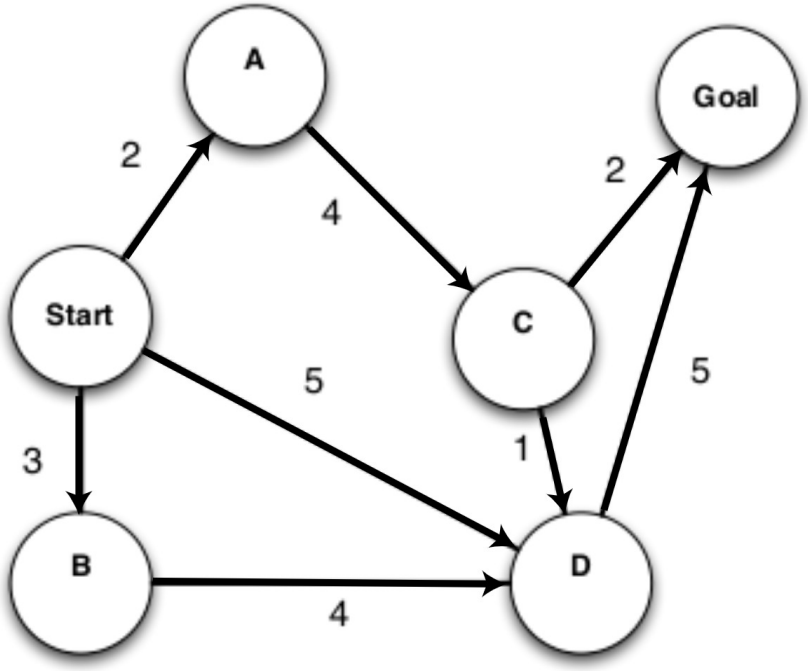
Review (Search Algorithms)

For each of the following search strategies, work out the path returned by the search on the graph shown above. In all cases, assume ties resolve in such a way that states with earlier alphabetical order are expanded first. The start and goal state are S and G, respectively.

a) Depth-first search.

b) Breadth-first search.

c) Uniform cost search.



Review (Search Algorithms)

- Which of the following are true and which are false? Explain your answers.

1. Depth-first search always expands at least as many nodes as A* search with an admissible heuristic.
2. h(n) = 0 is an admissible heuristic for the 8-puzzle.
3. Breadth-first search is complete even if zero step costs are allowed.
4. Assume that a rook can move on a chessboard any number of squares in a straight line, vertically or horizontally, but cannot jump over other pieces. Manhattan distance is an admissible heuristic for the problem of moving the rook from square A to square B in the smallest number of moves.

Review (Search Algorithms)

- The heuristic path algorithm is a best-first search in which the objective function is f(n) = (2−w)g(n) +w h(n). For what values of w is this algorithm guaranteed to be optimal? (You may assume tha this admissible.) What kind of search does this perform when w=0? When w=1? When w=2?