CS 730/730W/830: Intro AI

1 handout: slides

730W journal entries were due

EOLQs

Recap

Formalizing Problem Solving

State: hypothetical world state
Operators: actions that modify world
Goal: desired state or test

(Herbert Simon and Allen Newell, "Computer simulation of human thinking and problem solving", 1961)

Comparison

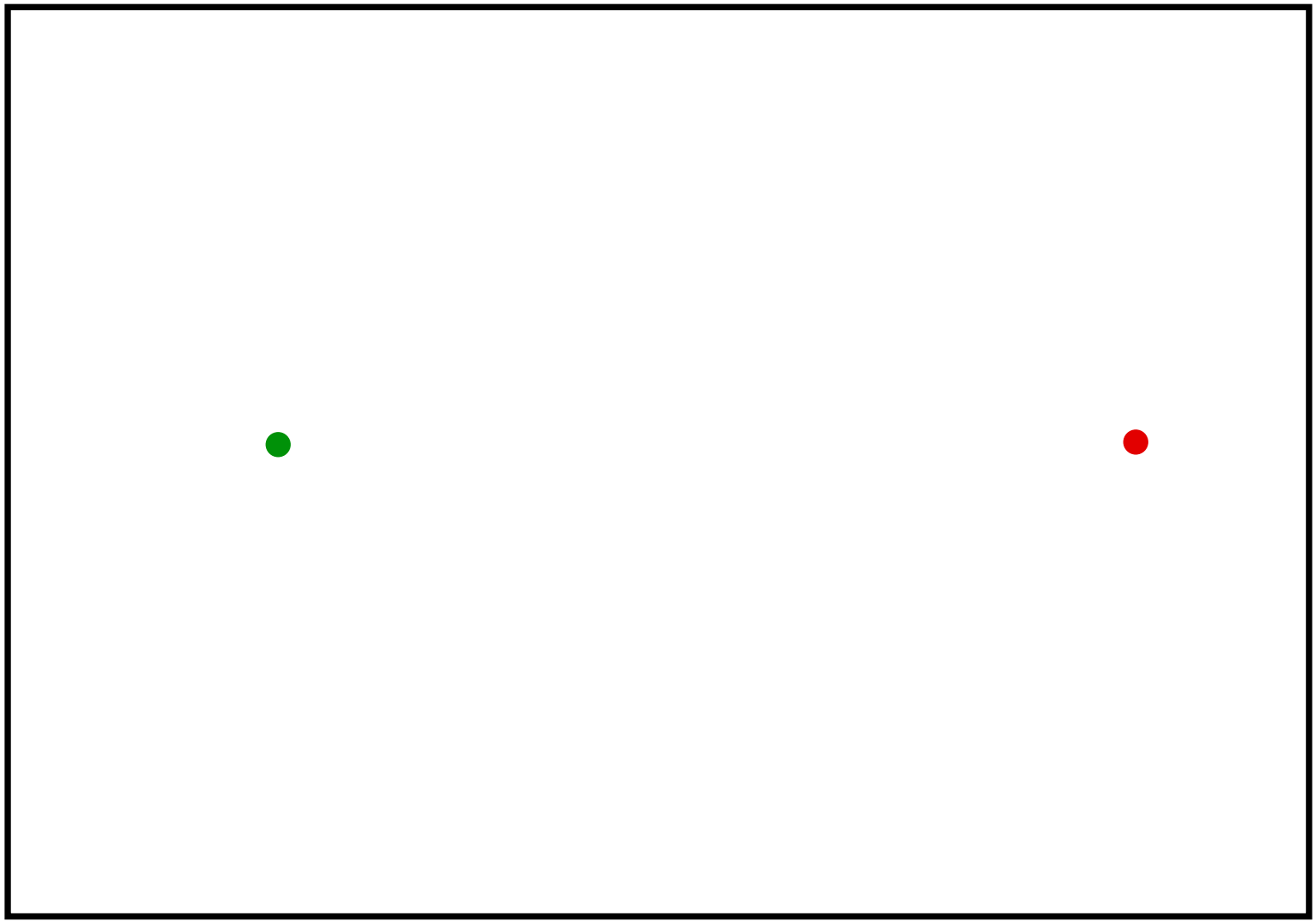
Algorithm	Time	Space	Complete	Admissible
Depth-first	b^m	bm	If $m \geq d$	No
Breadth-first	b^d	b^d	Yes	If ops cost 1
Uniform-cost	b^d	b^d	Yes	Yes
Iterative deepening	b^d	bd	Yes	Yes

branching factor b
maximum depth m
solution depth d

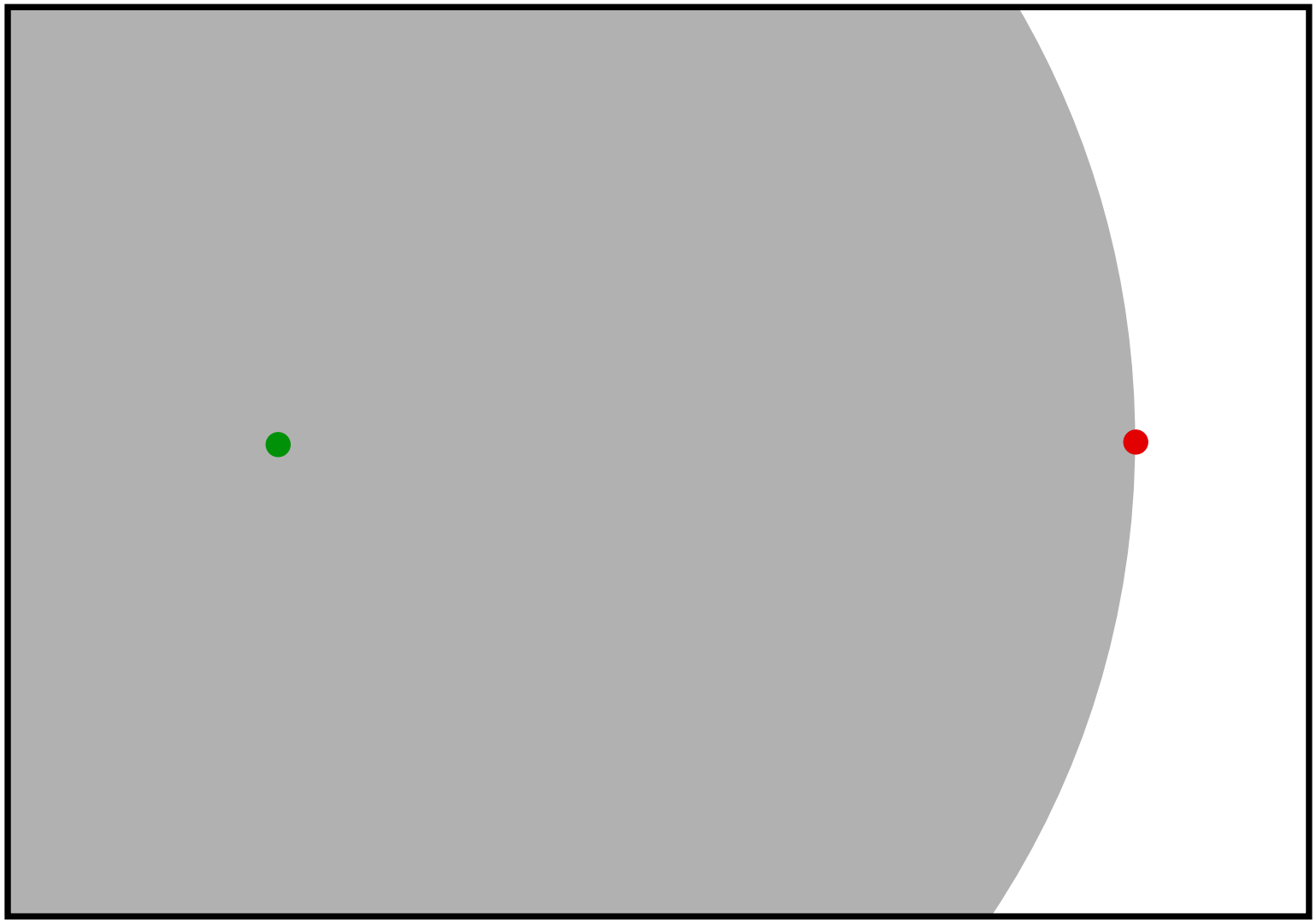
Are We Done?

Heuristic Search

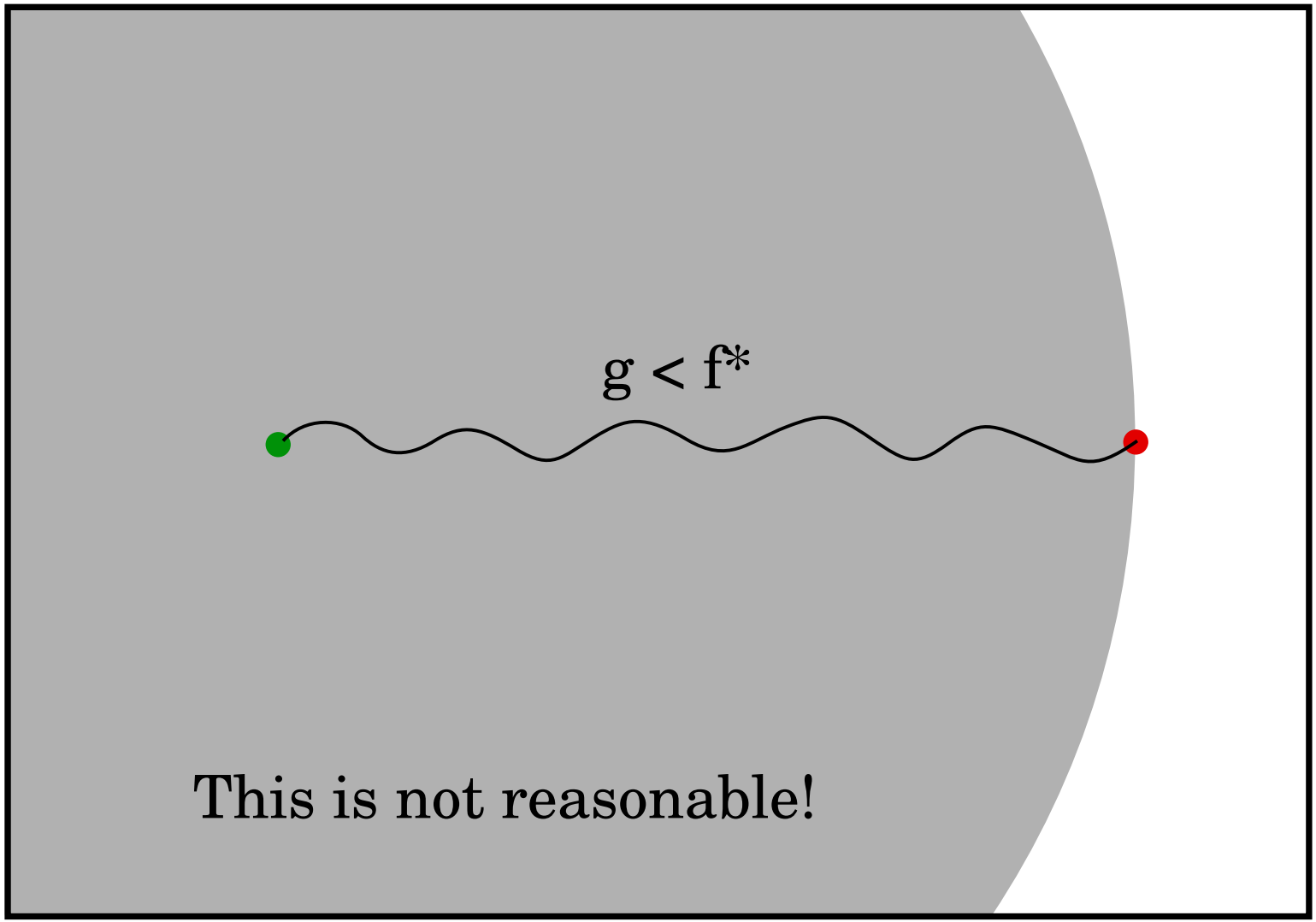
Dijkstra Behavior



Dijkstra Behavior



Dijkstra Behavior



Heuristic Evaluation

Heureka!
— *Archimedes*

Heuristic knowledge is useful, but not necessarily correct.

Heuristic algorithms use heuristic knowledge to solve a problem.

A *heuristic function* takes a state and returns an estimate of the distance to a goal.

(Newell and Ernst, 1965; Lin, 1965)

Greedy Search

$Q \leftarrow$ an ordered list containing just the initial state.
Loop
  If Q is empty,
    then return failure.
  Node $\leftarrow$ Pop(Q).
  If *Node* is a goal,
    then return *Node* (or path to it)
    else
      Children $\leftarrow$ Expand (*Node*).
      Merge *Children* into Q, keeping **sorted by heuristic**.$\leftarrow$

Greedy on the 8-puzzle

$h(n) =$ number of tiles out of place. (The blank is not a tile.)

	2	8	3		1	2	3
Start state:	1	6	4	Goal state:	8	␣	4
	7	␣	5		7	6	5

Please draw the tree resulting from the first two node expansions.

Evaluating Greedy

Assume branching factor b and solution at depth d.

Completeness:

Time:

Space:

Admissibility:

Break

- slides: handout, website
- recitation
- blog entries due on Wednesdays, 8am
- asst 1
  - duplicates
  - make script, submission
  - transcript but no hardcopy for milestone
  - reference solution
  - standard libraries
- office hours
- projects (Apr 2), UROP (Mar 1), UNH AI group (see wiki)

A* Search

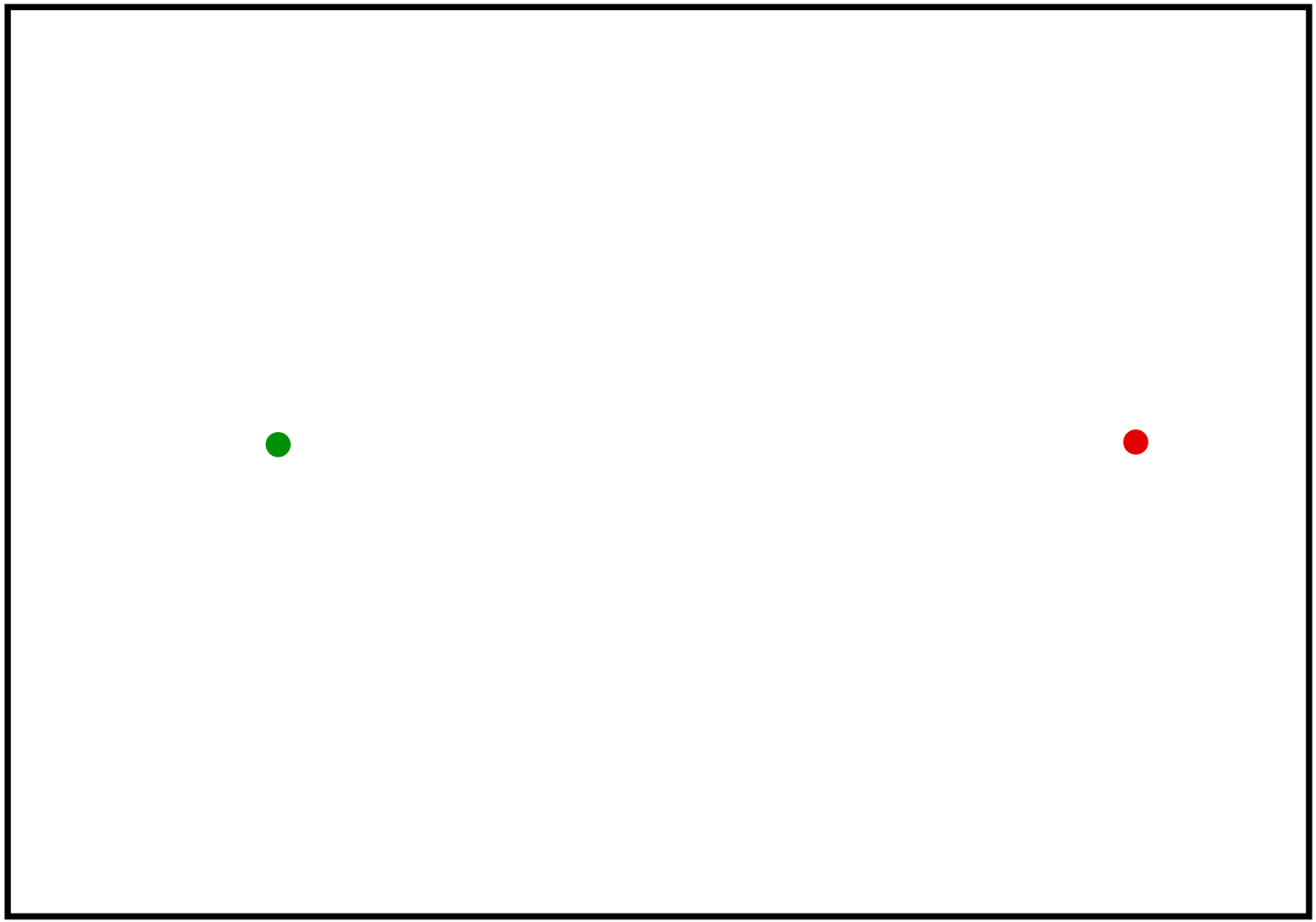
Consider estimated final path cost! $f(n) = g(n) + h(n)$

$Q \leftarrow$ an ordered list containing just the initial state.
Loop
  If Q is empty,
    then return failure.
  Node $\leftarrow$ Pop(Q).
  If *Node* is a goal,
    then return *Node* (or path to it)
    else
      Children $\leftarrow$ Expand (*Node*).
      Merge *Children* into Q, keeping **sorted by** $f(n)$. $\leftarrow$

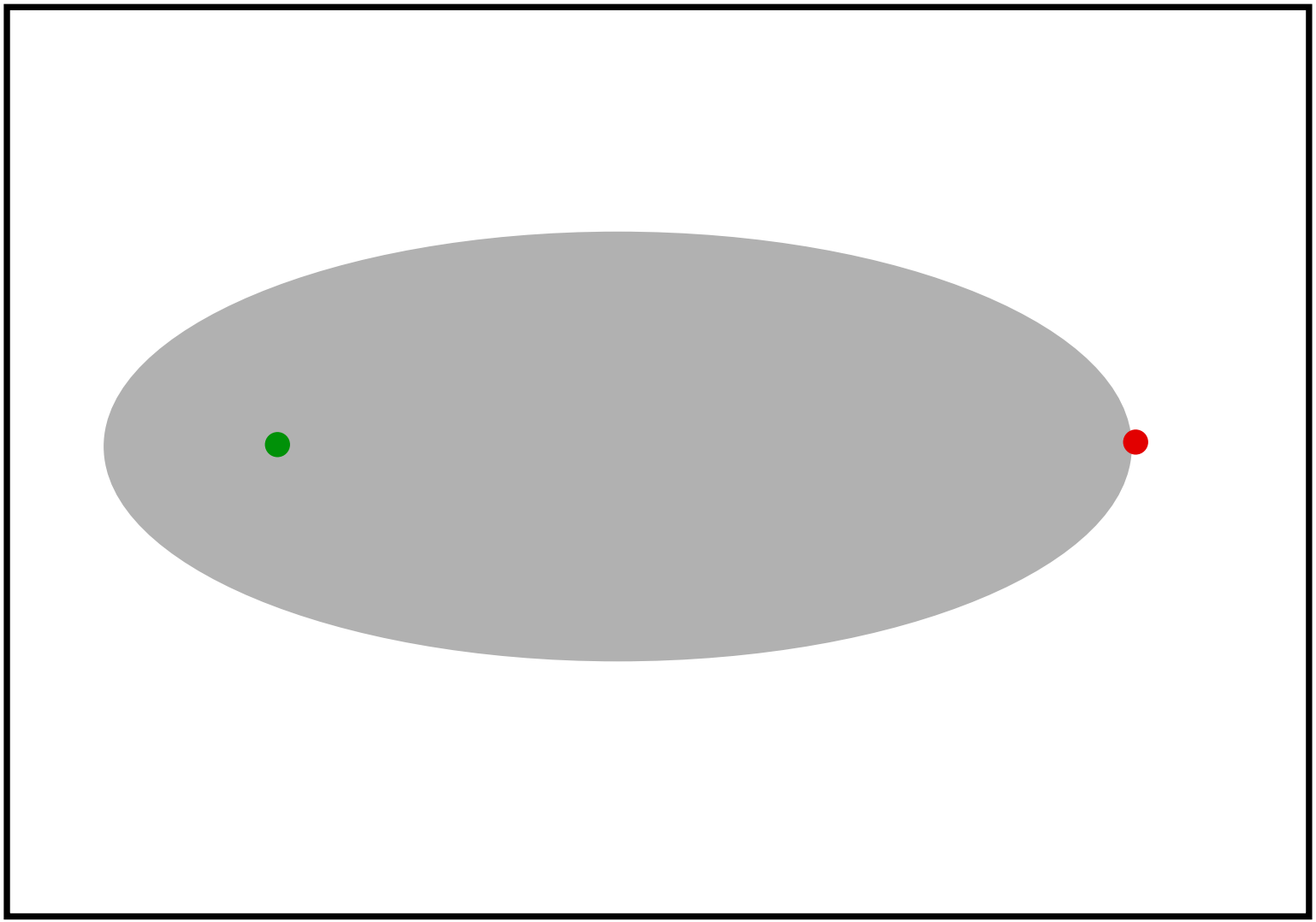
A* on the 8-puzzle

A* doesn't get sidetracked like best-first.

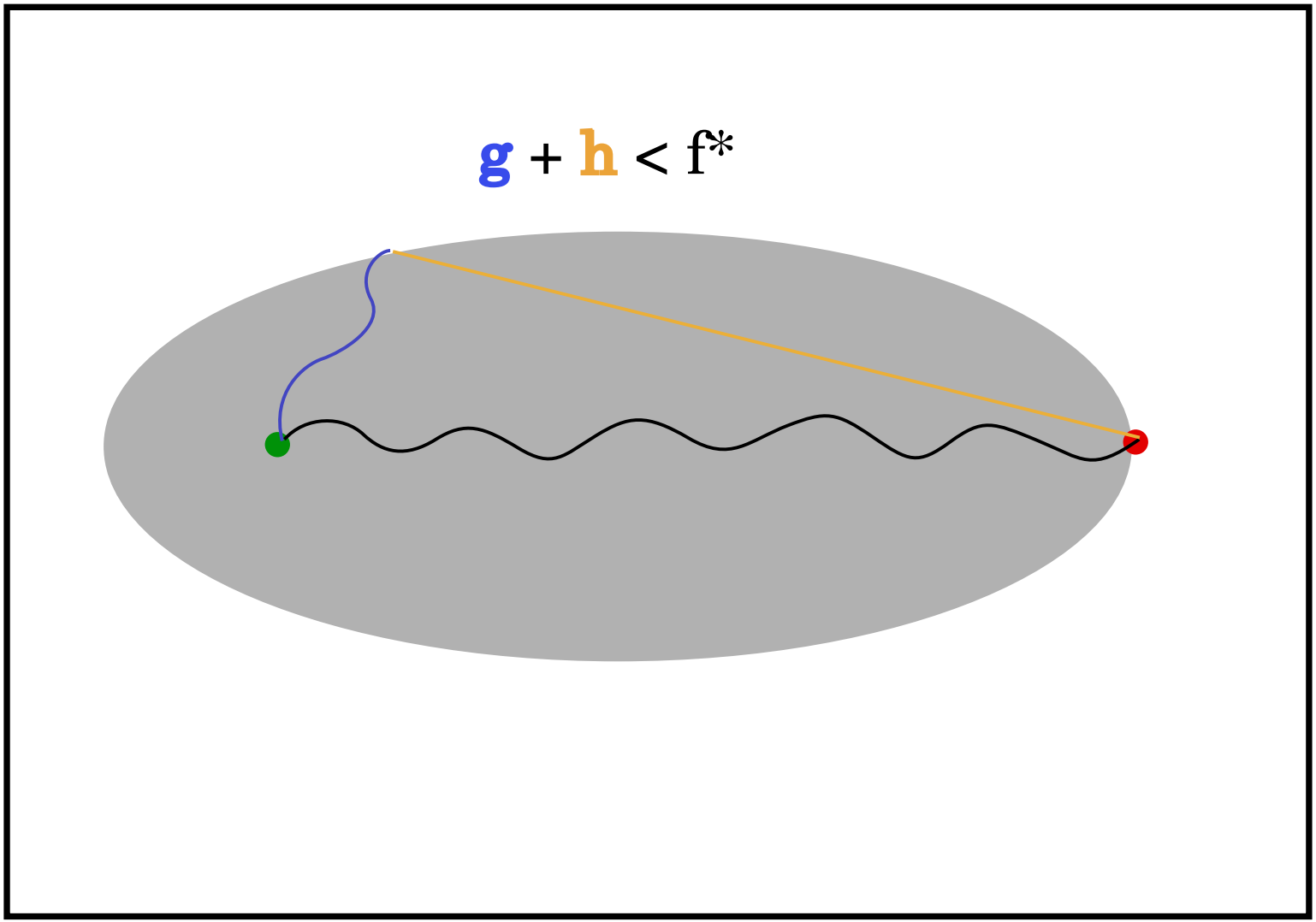
Why Fewer Nodes?



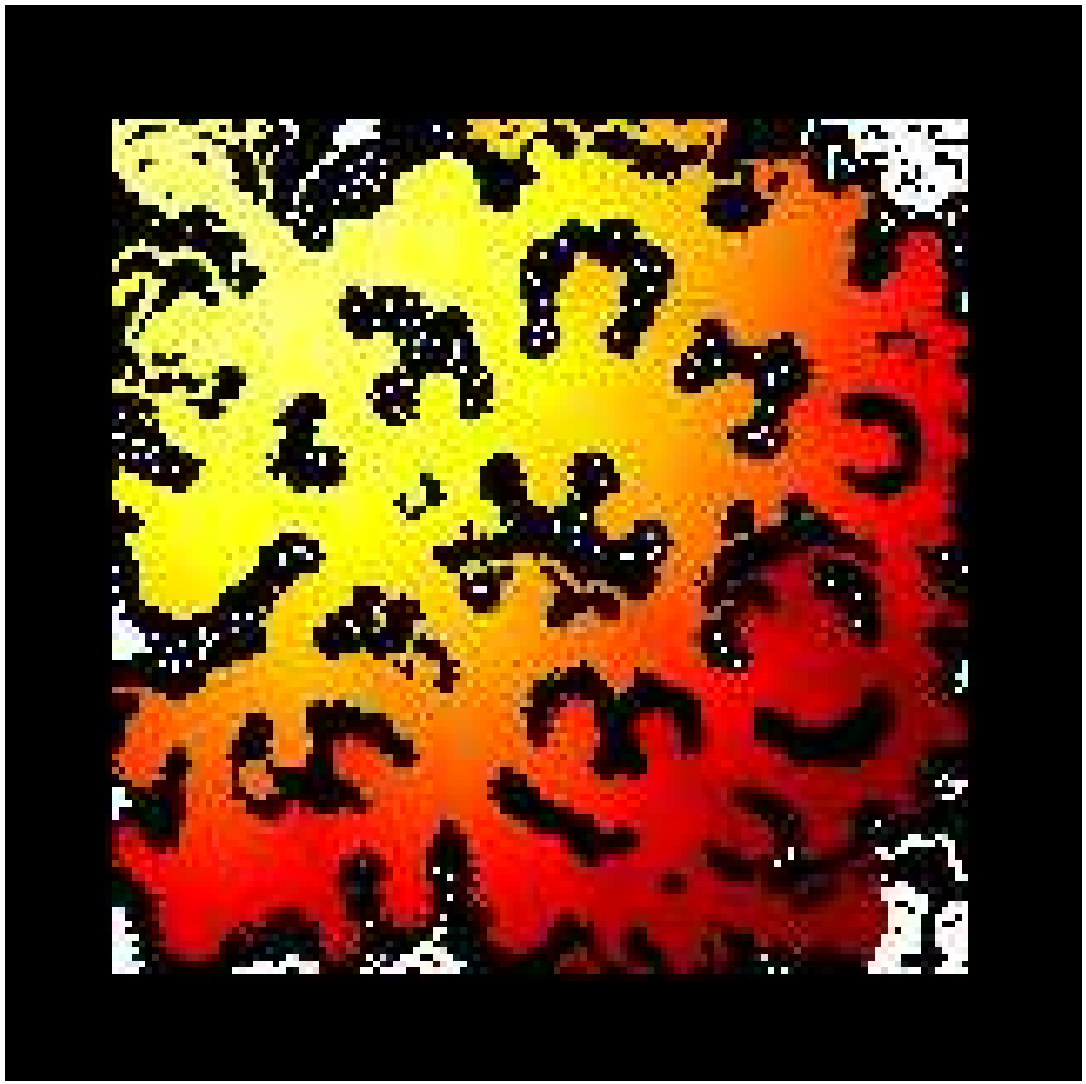
Why Fewer Nodes?



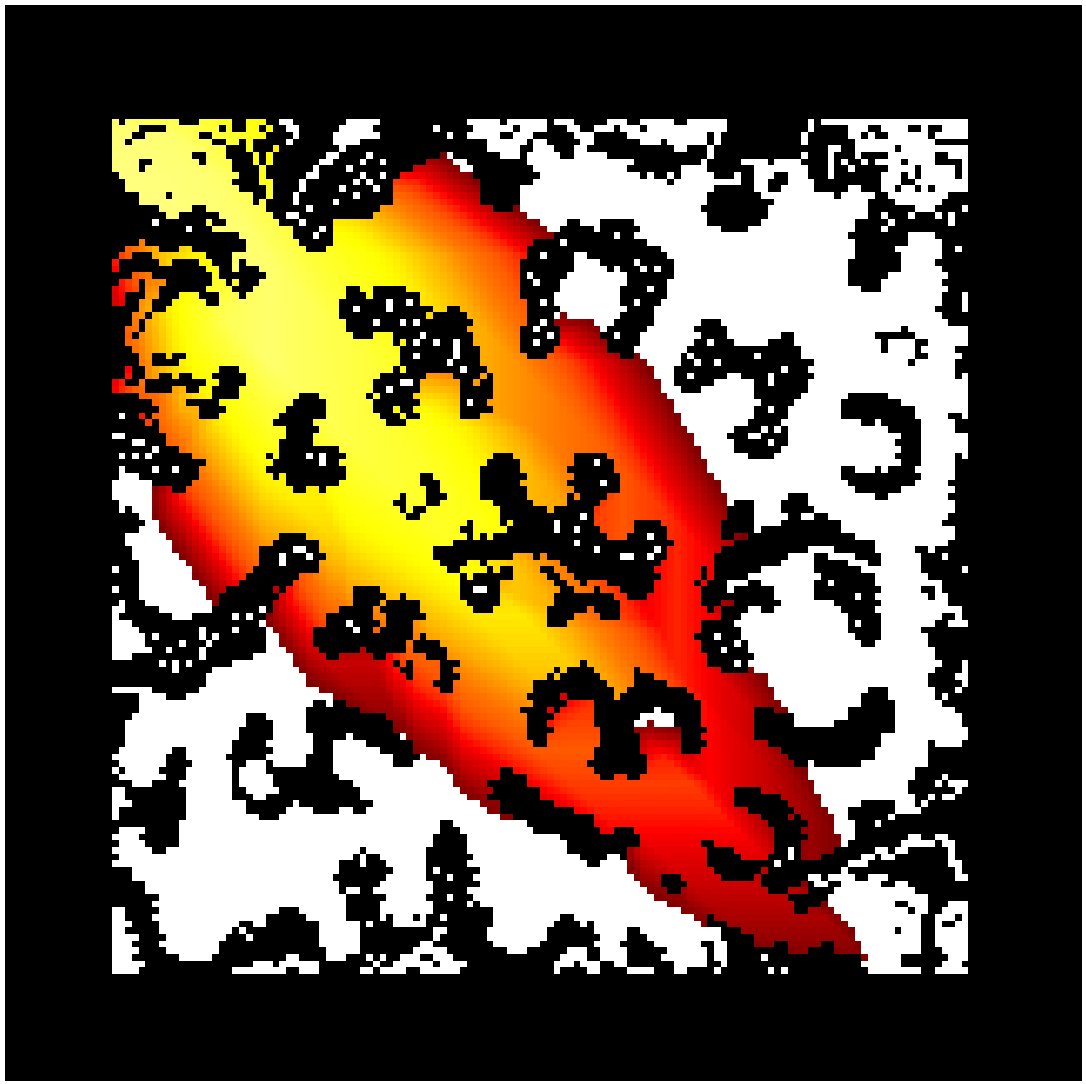
Why Fewer Nodes?



Uniform-cost Search Behavior



A* Behavior



Evaluating A*

Assume branching factor b and solution at depth d.

Completeness:

Time:

Space:

Admissibility:

Admissibility of A*

Optimality of A*

1. For admissible h, f can be made non-decreasing.
2. A* expands nodes in order of non-decreasing f.
3. Must examine all nodes with $f < f^*$.

Heuristic Functions

1. Find by relaxing problem
2. Want cheap computation
3. Want highest value
   (a) If $h_1(n) \leq h_2(n)$ for all n, h_2 dominates h_1

Beam Search

Truncate queue to hold k nodes.

k is the *beam width*.

Can be used with any queue ordering (such as $g(n)$ or $f(n)$).

Other Shortest-path Algorithms

- IDA*
- SMA*, IE
- RBFS
- RTA*, LRTA*
- A^*_ϵ

Course projects!

Search Algorithms

- Uninformed methods
  - Depth-first
  - Breadth-first
  - Uniform-cost
- Informed methods ('best-first')
  - Greedy
  - A*
- Bounding memory
  - Iterative deepening
  - Beam search

EOLQs

Please write down the most pressing question you have about the course material covered so far and put it in the box on your way out.

Thanks!