1 handout: slides
asst 2 on-line

# 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)

Is iterative deepening real?

# 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 |
| IDDFS | $b^d(\frac{b}{b-1})^2$ | $bd$ | Yes | Yes |

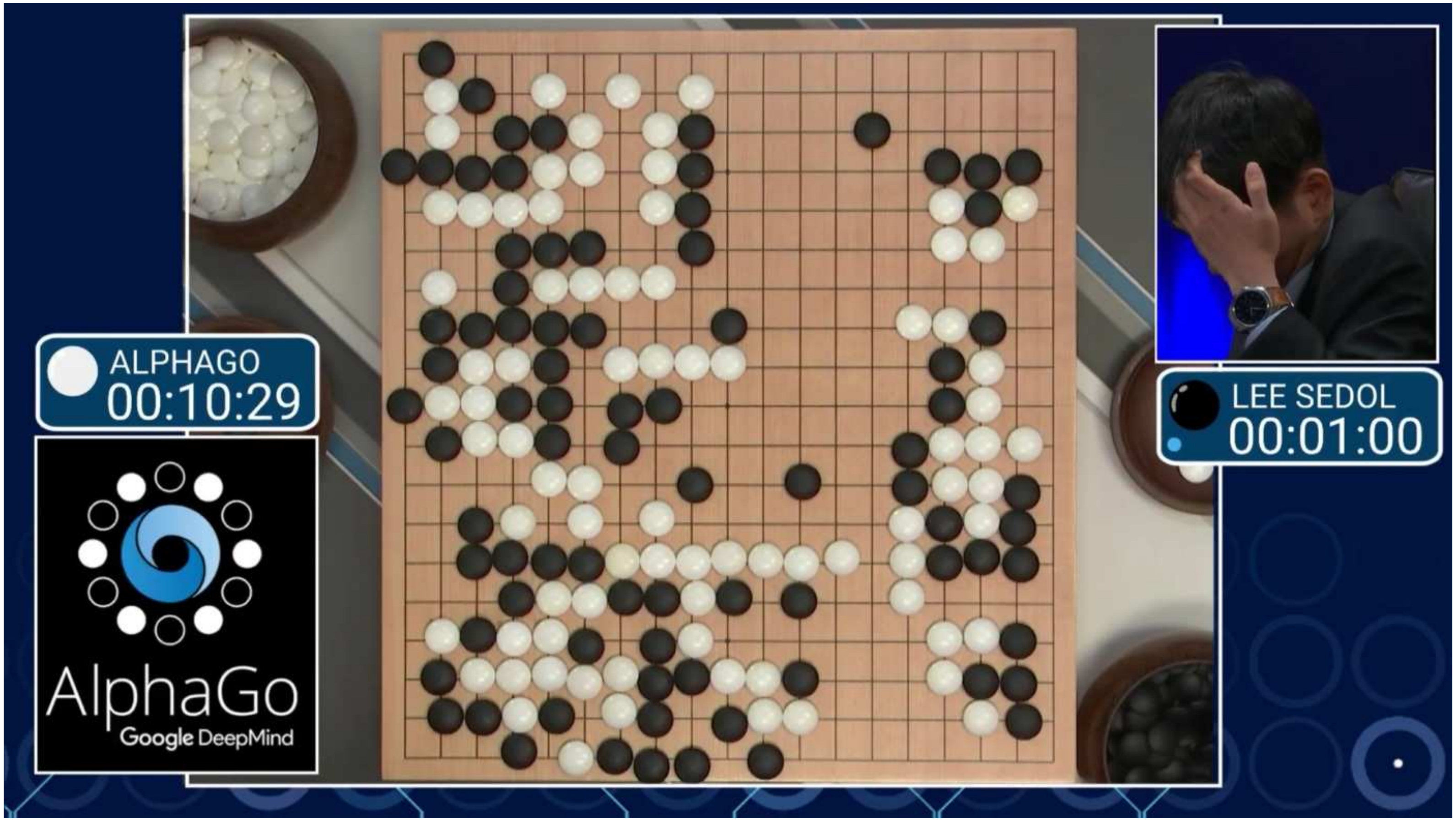
| | |
|---|---|
| branching factor | b |
| maximum depth | m |
| solution depth | d |

# Are We Done?



# Are We Done?

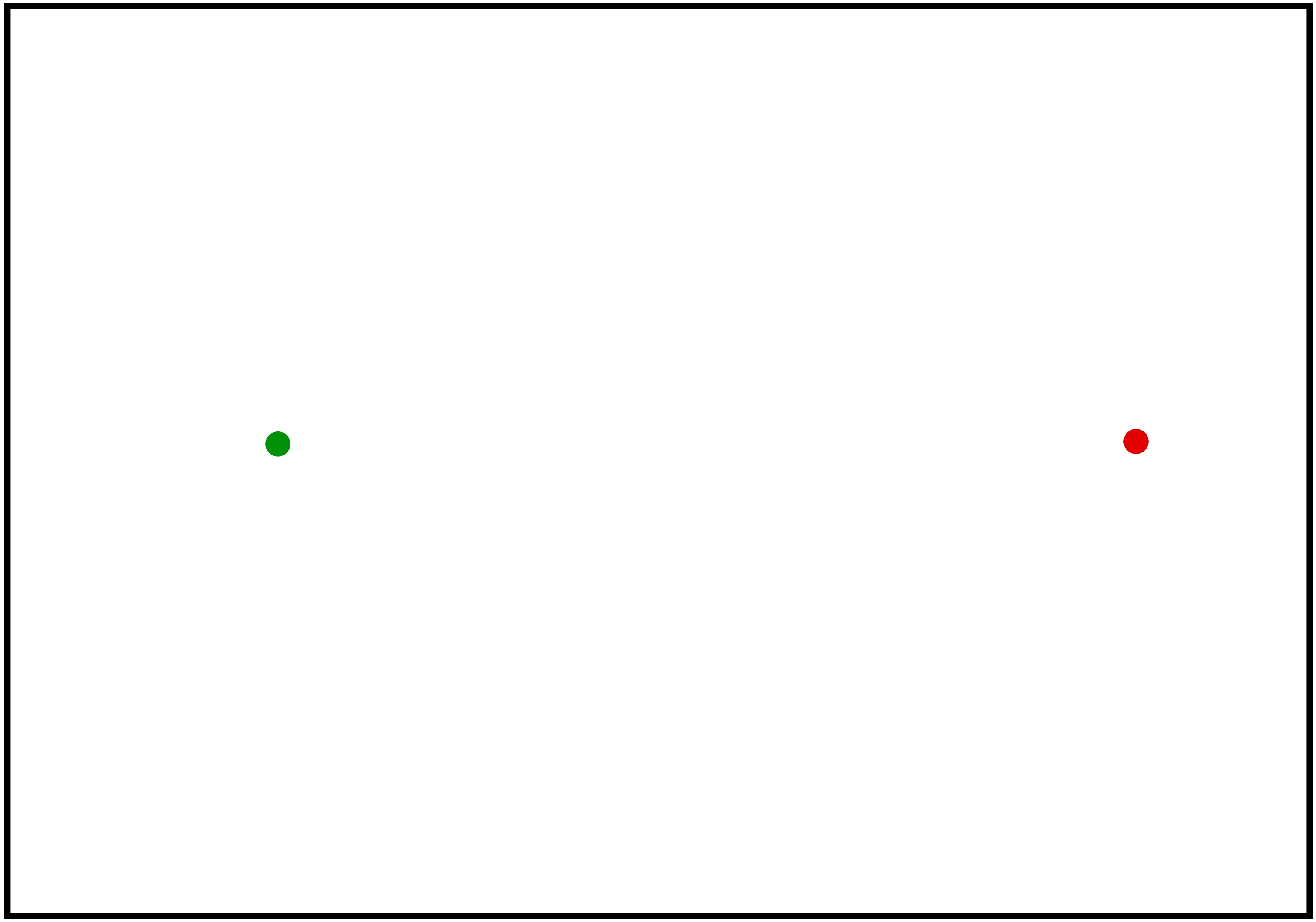




# Heuristic Search

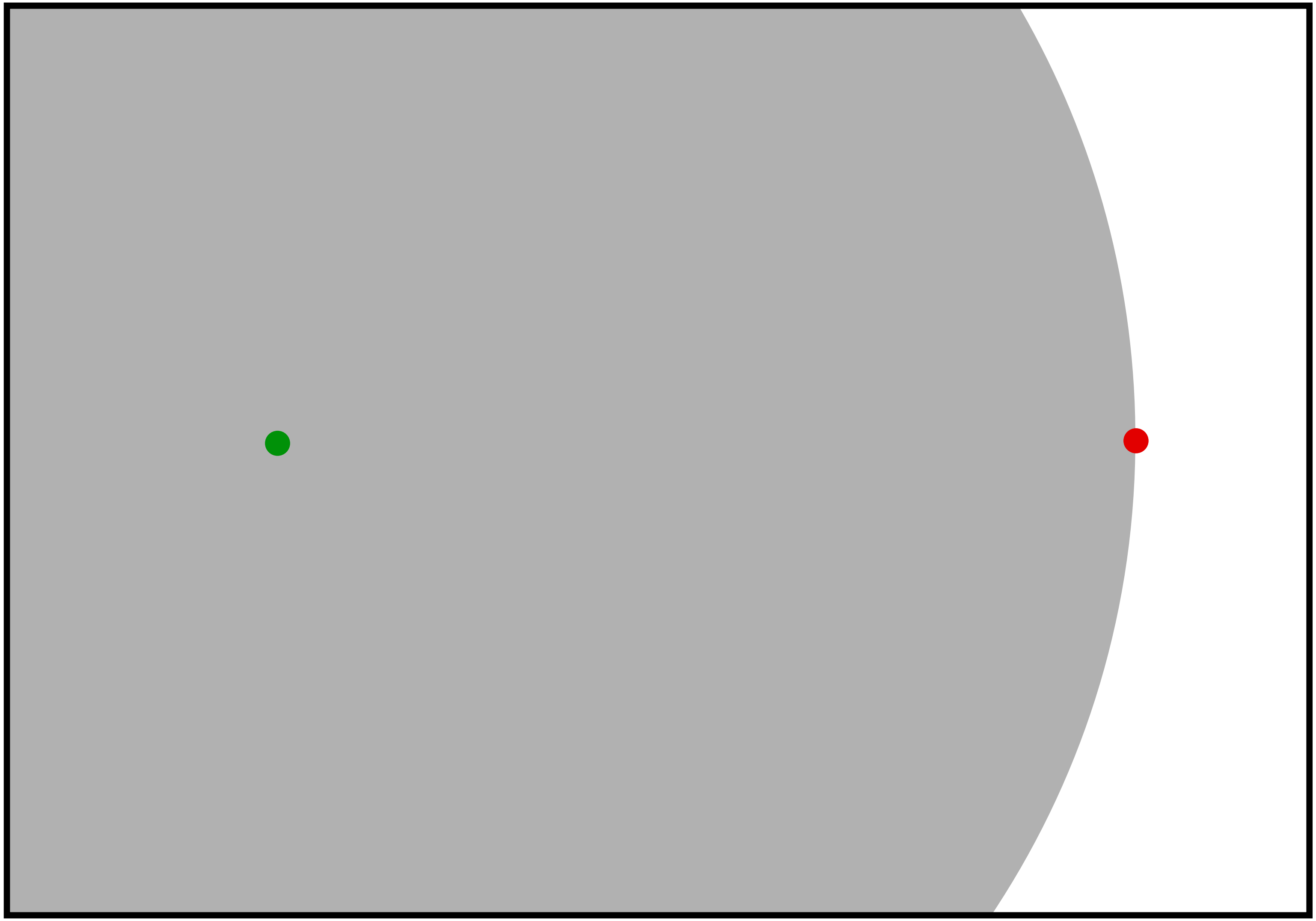
# UCS Behavior





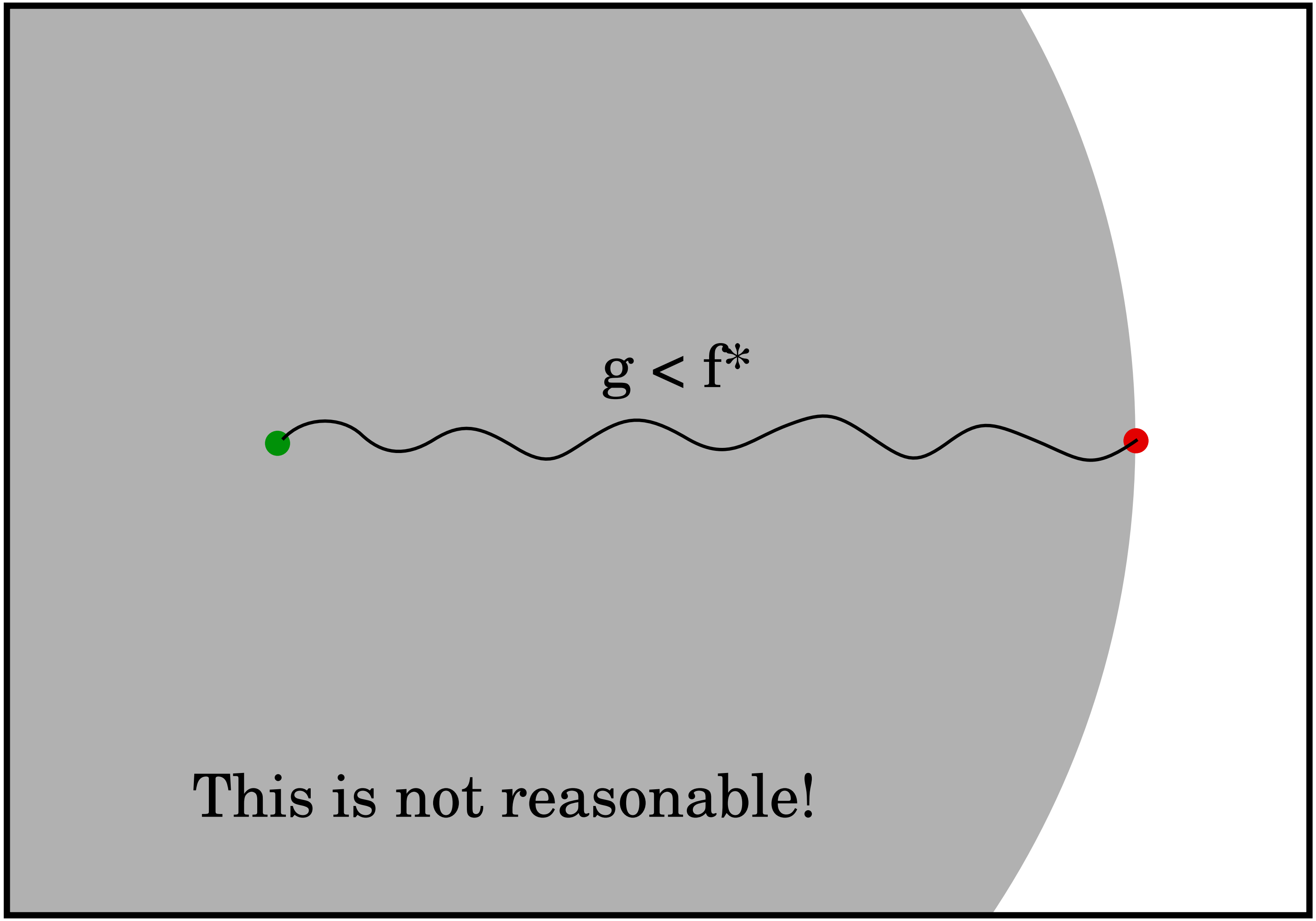
# UCS Behavior





# UCS 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 a lower bound on the cost-to-go to reach a goal.

(Newell and Ernst, 1965; Lin, 1965)

# 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** $f(n)$.   $\leftarrow$

## An Example: the 8-puzzle



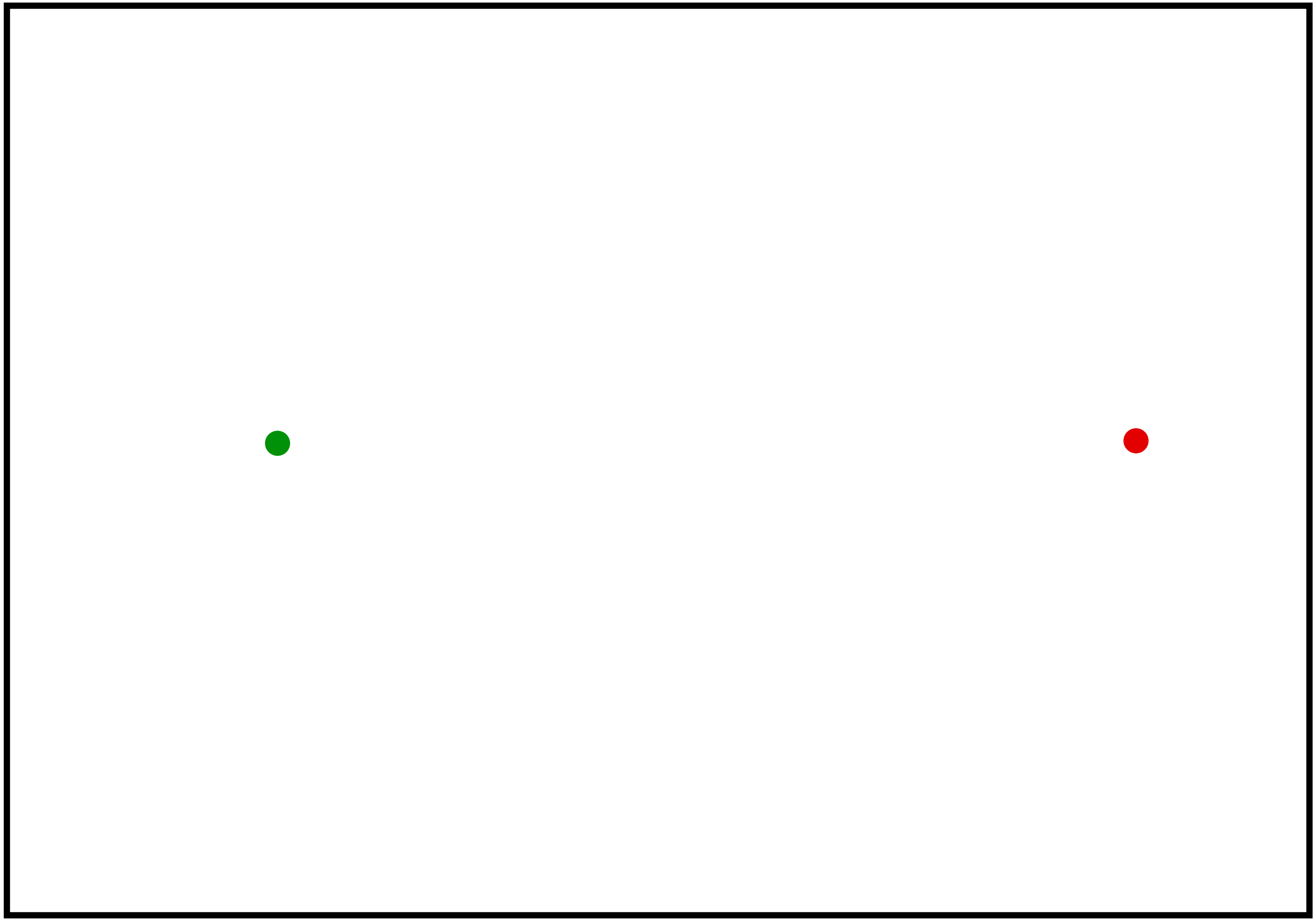
$h(n) =$ number of tiles out of place. (The blank is not a tile.)

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | 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.

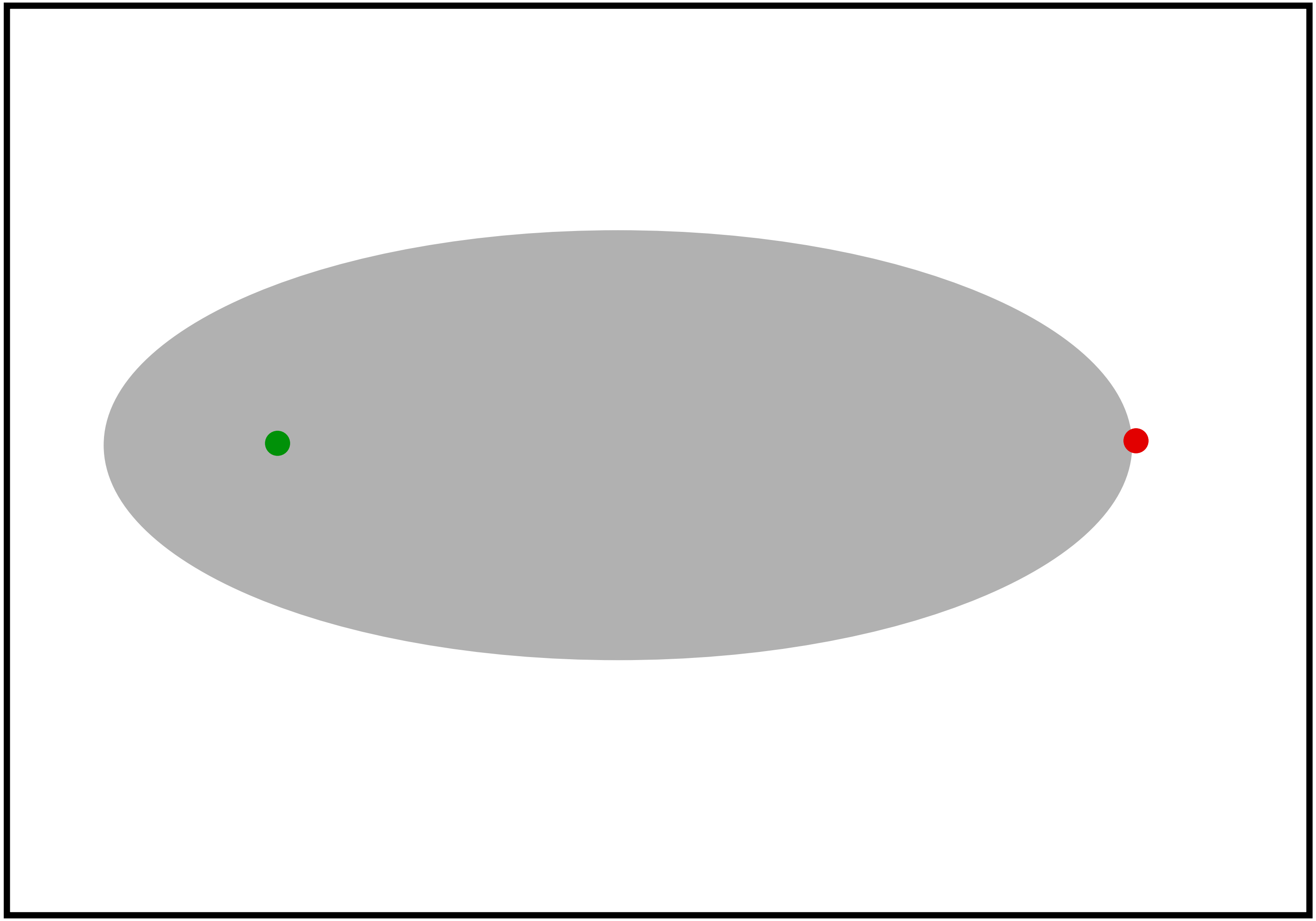
# Why Fewer Nodes?





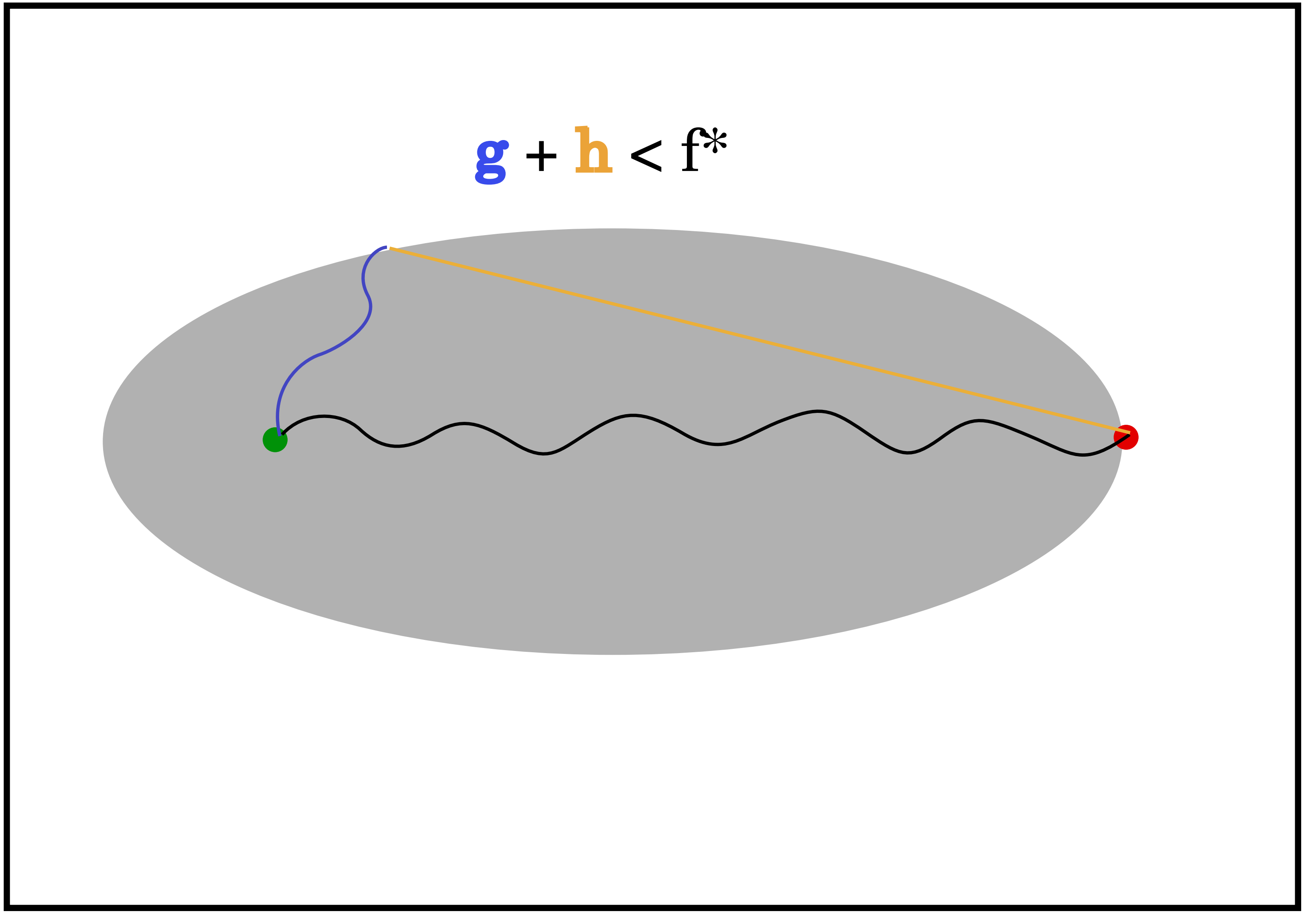
# Why Fewer Nodes?





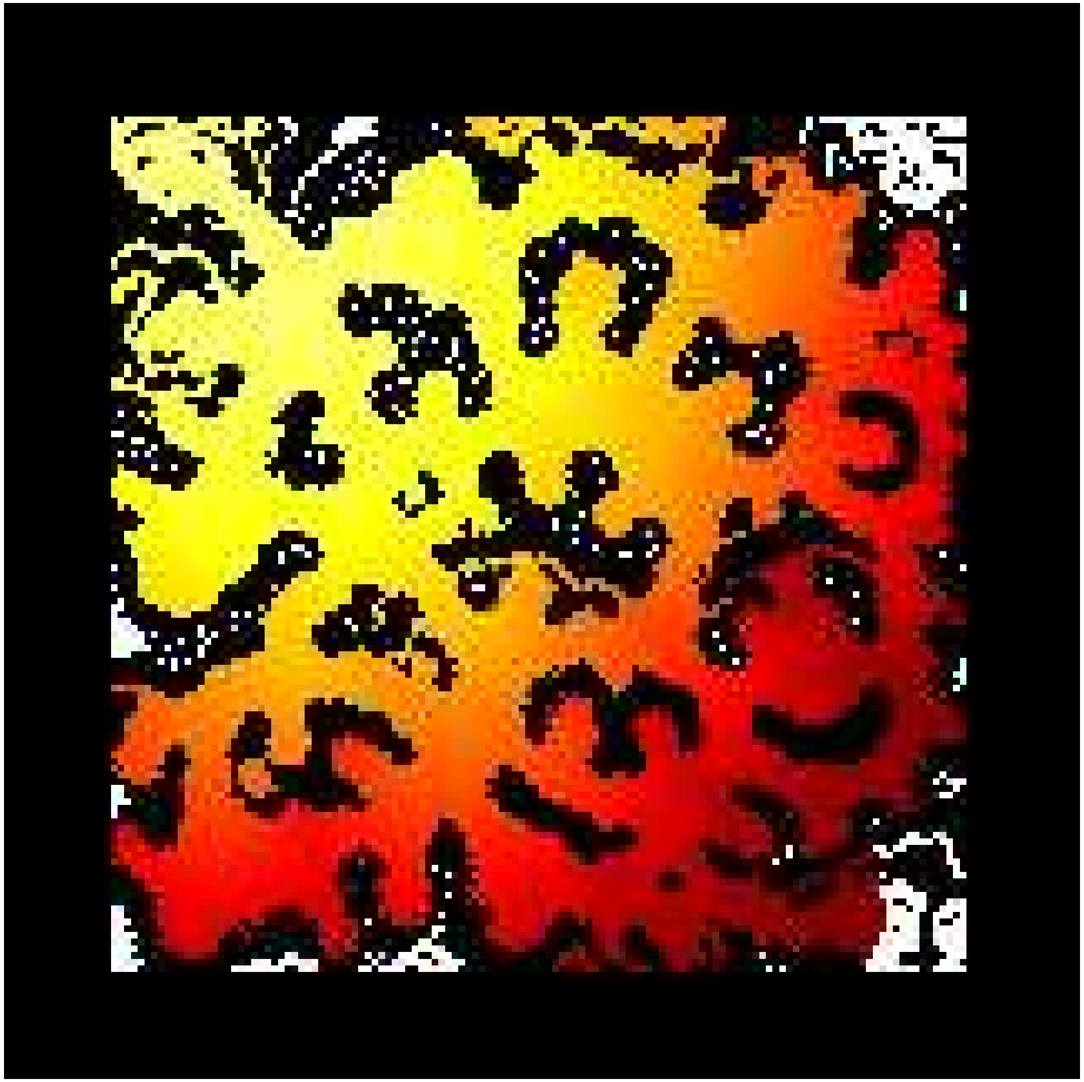
# Why Fewer Nodes?





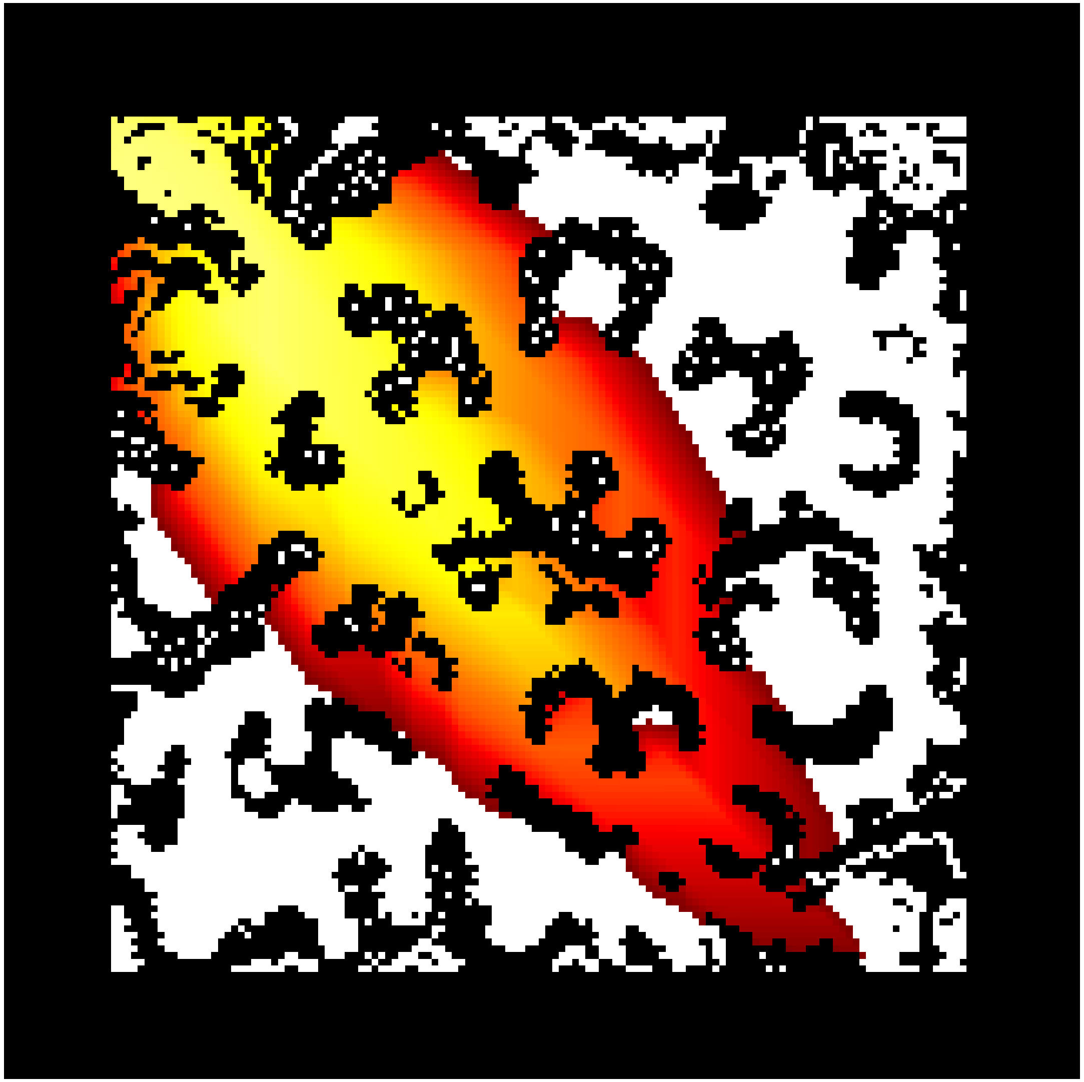
# UCS Behavior





# A* Behavior





# Break



- asst 1
- asst 2
- collaboration
- Google 'UNH AI group wiki'
- projects!

# Evaluating A*



Assume branching factor $b$ and solution at depth $d$.

Completeness:
Time:
Space:
Admissibility:

# Admissibility of A*



If $h$ is admissible, other solutions at least as expensive!

1. Suppose $p$ is on path to better solution $b$.
2. $f(p) = g(p) + h(p) \leq f(b) = g(b)$
3. But $g(s) = f(s) \leq f(p)$.
4. So $g(s) \leq g(b)$.

# 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 $f < f^*$.

# Heuristics



Simplified problem must give lower bound on original!

1. Relaxation: fewer and/or weaker constraints
   - Sometime efficient closed form
2. Abstraction: simplify token identity
   - Smaller search space

Want highest value

- If $h_1(n) \leq h_2(n)$ for all $n$, $h_2$ dominates $h_1$

Need fast computation

# Admissibility



how to prove heuristic $h$ admissible for problem $P$:

1. identify a simplified problem $P'$
2. show $h$ is optimal for $P'$
3. show that $P'$ easier than $P$

   optimal solution for instance $i'$ of $P'$ is $\leq$
   optimal sol for instance $i$ of $P$
4. since $h$ gives optimal values for $P'$, it is therefore admissible for $P$

# 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!*