Recap
Heuristic Search

1 handout: slides
Recap

- Problem Solving
- DeGroot Results
- Comparison
- Are We Done?

Heuristic Search
Formalizing Problem Solving

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

Is iterative deepening real?
<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Time</th>
<th>Space</th>
<th>Complete</th>
<th>Admissible</th>
</tr>
</thead>
<tbody>
<tr>
<td>Depth-first</td>
<td>$b^m$</td>
<td>$b^m$</td>
<td>If $m \geq d$</td>
<td>No</td>
</tr>
<tr>
<td>Breadth-first</td>
<td>$b^d$</td>
<td>$b^d$</td>
<td>Yes</td>
<td>If ops cost 1</td>
</tr>
<tr>
<td>Uniform-cost</td>
<td>$b^d$</td>
<td>$b^d$</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>IDDFS</td>
<td>$b^d$</td>
<td>$bd$</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

branching factor $b$
maximum depth $m$
solution depth $d$
Are We Done?

Recap
- Problem Solving
- DeGroot Results
- Comparison
- Are We Done?

Heuristic Search
Heuristic Search

Recap

Heuristic Search
- Dijkstra Behavior
- Terminology
- GBFS
- Greed on 8-puzzle
- Evaluating Greedy
- Break
- A* Search
- A* on 8-puzzle
- Why Fewer Nodes?
- Uniform-cost Search Behavior
- A* Behavior
- Evaluating A*
- Admissibility
- Optimality of A*
- Heuristics
- Search Algorithms
- EOLQs

Wheeler Ruml (UNH)
Dijkstra Behavior

Recap
Heuristic Search
- Dijkstra Behavior
  - Terminology
  - GBFS
  - Greed on 8-puzzle
  - Evaluating Greedy
  - Break
  - A* Search
  - A* on 8-puzzle
  - Why Fewer Nodes?
- Uniform-cost Search Behavior
- A* Behavior
- Evaluating A*
- Admissibility
- Optimality of A*
- Heuristics
- Search Algorithms
- EOLQs
Dijkstra Behavior

Recap

Heuristic Search

- Dijkstra Behavior
  - Terminology
  - GBFS
  - Greed on 8-puzzle
  - Evaluating Greedy
  - Break
  - A* Search
  - A* on 8-puzzle
  - Why Fewer Nodes?
  - Uniform-cost
- Search Behavior
  - A* Behavior
  - Evaluating A*
  - Admissibility
  - Optimality of A*
  - Heuristics
  - Search Algorithms
  - EOLQs

Wheeler Ruml (UNH)
Dijkstra Behavior

Recap
Heuristic Search
- Dijkstra Behavior
  - Terminology
  - GBFS
  - Greed on 8-puzzle
  - Evaluating Greedy
  - Break
  - A* Search
  - A* on 8-puzzle
  - Why Fewer Nodes?
  - Uniform-cost
  - Search Behavior
  - A* Behavior
  - Evaluating A*
  - Admissibility
  - Optimality of A*
  - Heuristics
  - Search Algorithms
  - EOLQs

This is not reasonable!
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)
Greedy Best-first Search (BGFS)

Recap

Heuristic Search
- Dijkstra Behavior
- Terminology
- GBFS
- Greed on 8-puzzle
- Evaluating Greedy
- Break
- A* Search
- A* on 8-puzzle
- Why Fewer Nodes?
- Uniform-cost Search Behavior
- A* Behavior
- Evaluating A*
- Admissibility
- Optimality of A*
- Heuristics
- Search Algorithms
- EOLQs

\( Q \leftarrow \text{an ordered list containing just the initial state.} \)

Loop

If \( Q \) is empty,

then return failure.

\( \text{Node} \leftarrow \text{Pop}(Q). \)

If \( \text{Node} \) is a goal,

then return \( \text{Node} \) (or path to it)

else

\( \text{Children} \leftarrow \text{Expand}(\text{Node}). \)

Merge \( \text{Children} \) into \( Q \), keeping sorted by heuristic.
Greedy on the 8-puzzle

\[ h(n) = \text{number of tiles out of place. (The blank is not a tile.)} \]

\[
\begin{array}{ccc}
2 & 8 & 3 \\
7 & \square & 5 \\
\end{array}
\quad \quad \begin{array}{ccc}
1 & 2 & 3 \\
8 & \square & 4 \\
7 & 6 & 5 \\
\end{array}
\]

Start state: 1 6 4  Goal state: 8 \[ \square \] 4

Please draw the tree resulting from the first two node expansions.
Assume branching factor $b$ and solution at depth $d$.

**Completeness:**

**Time:**

**Space:**

**Admissibility:**
Recap

Heuristic Search
- Dijkstra Behavior
- Terminology
- GBFS
- Greed on 8-puzzle
- Evaluating Greedy
- Break
- A* Search
- A* on 8-puzzle
- Why Fewer Nodes?
- Uniform-cost Search Behavior
- A* Behavior
- Evaluating A*
- Admissibility
- Optimality of A*
- Heuristics
- Search Algorithms
- EOLQs

- asst 1
- asst 2
- office hours
- UNH AI group: Wednesdays at 12:30pm in Kingsbury N233
- projects!
- UROP
Consider estimated final path cost! \( f(n) = g(n) + h(n) \)

\[ Q \leftarrow \text{an ordered list containing just the initial state.} \]

Loop

If \( Q \) is empty,
then return failure.

\( \text{Node} \leftarrow \text{Pop}(Q) \).

If \( \text{Node} \) is a goal,
then return \( \text{Node} \) (or path to it)
else

\( \text{Children} \leftarrow \text{Expand}(\text{Node}). \)

Merge \( \text{Children} \) into \( Q \), keeping sorted by \( f(n) \).
A* doesn’t get sidetracked like best-first.
Why Fewer Nodes?

Recap
Heuristic Search
- Dijkstra Behavior
- Terminology
- GBFS
- Greed on 8-puzzle
- Evaluating Greedy
- Break
- A* Search
- A* on 8-puzzle
- Why Fewer Nodes?

- Uniform-cost Search Behavior
- A* Behavior
- Evaluating A*
- Admissibility
- Optimality of A*
- Heuristics
- Search Algorithms
- EOLQs
Why Fewer Nodes?

Recap
- Heuristic Search
  - Dijkstra Behavior
  - Terminology
  - GBFS
  - Greed on 8-puzzle
  - Evaluating Greedy
  - Break
  - A* Search
  - A* on 8-puzzle

- Uniform-cost Search Behavior
- A* Behavior
- Evaluating A*
- Admissibility
- Optimality of A*
- Heuristics
- Search Algorithms
- EOLQs
Why Fewer Nodes?

Recap
Heuristic Search
- Dijkstra Behavior
- Terminology
- GBFS
- Greed on 8-puzzle
- Evaluating Greedy
- Break
- A* Search
- A* on 8-puzzle
- Why Fewer Nodes?
  - Uniform-cost Search Behavior
  - A* Behavior
  - Evaluating A*
  - Admissibility
  - Optimality of A*
  - Heuristics
  - Search Algorithms
  - EOLQs
Uniform-cost Search Behavior

Recap

Heuristic Search
- Dijkstra Behavior
- Terminology
- GBFS
- Greed on 8-puzzle
- Evaluating Greedy
- Break
- A* Search
- A* on 8-puzzle
- Why Fewer Nodes?
- Uniform-cost Search Behavior
- A* Behavior
- Evaluating A*
- Admissibility
- Optimality of A*
- Heuristics
- Search Algorithms
- EOLQs
Assume branching factor $b$ and solution at depth $d$. 

Completeness:

Time:

Space:

Admissibility:
If $h$ is admissible, nodes leading to a better solution can’t exist!
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

Recap

Heuristic Search
- Dijkstra Behavior
- Terminology
- GBFS
- Greed on 8-puzzle
- Evaluating Greedy
- Break
- A* Search
- A* on 8-puzzle
- Why Fewer Nodes?
- Uniform-cost Search Behavior
- A* Behavior
- Evaluating A*
- Admissibility
- Optimality of A*

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
## Search Algorithms

### Recap

- Dijkstra Behavior
- Terminology
- GBFS
- Greed on 8-puzzle
- Evaluating Greedy
- Break
- A* Search
- A* on 8-puzzle
- Why Fewer Nodes?
- Uniform-cost
- Search Behavior
- A* Behavior
- Evaluating A*
- Admissibility
- Optimality of A*
- Heuristics

### Heuristic Search

- **Uninformed methods**
  - Depth-first
  - Breadth-first
  - Uniform-cost

- **Informed methods ('best-first')**
  - GBFS
  - A*

- **Bounding memory**
  - Iterative deepening
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!