Search Algorithms: Plan

- Environments as search spaces
- Properties of search algorithms
- Uninformed search algorithms
  - Breadth-first search
  - Depth-first search
Search Spaces

- Initial state
- Actions, transitions
- Goal state(s)
- Transition costs

Problem Description

- Search tree/graph
- Expanding a node
- Frontier
Properties of Search Algorithms

• Completeness: algo will find a goal state if one exists
• Optimality: algo will find the least-cost path to goal state
• Time complexity: number of operations needed to find goal state
• Space complexity: amount of memory needed to find goal state
• Complexity usually expressed in terms of depth of goal and branching factor
function TREE-SEARCH(problem) returns a solution, or failure

initialize the frontier using the initial state of problem

repeat:

    if the frontier is empty then return failure

    extract a node from the frontier

    if the node contains a goal state then return the solution

    expand the chosen node, adding the resulting nodes to frontier

(FIFO: Breadth-first, LIFO: Depth-first)
Additional Variants

• Depth-limited search:
  • Run DFS, but don’t search deeper than depth limit $L$

• Iterative-deepening:
  • Run depth-limited search with depth $L = 1$, then $L = 2$, …
  • Until goal state found

• Uniform-cost search:
  • Variation of breadth-first. Expand nodes in order of cost from initial state.
## Properties

<table>
<thead>
<tr>
<th></th>
<th>Complete</th>
<th>Optimal</th>
<th>Time</th>
<th>Space</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Breadth-First</strong></td>
<td>Yes</td>
<td>Yes if uniform cost, no otherwise</td>
<td>$O(b^d)$</td>
<td>$O(b^d)$</td>
</tr>
<tr>
<td><strong>Depth-First</strong></td>
<td>Yes if and only if (iff) finite</td>
<td>No</td>
<td>$O(b^{\text{max_depth}})$</td>
<td>$O(b(\text{max_depth}))$</td>
</tr>
<tr>
<td><strong>Depth-Limited</strong></td>
<td>No</td>
<td>No</td>
<td>$O(b^L)$</td>
<td>$O(bL)$</td>
</tr>
<tr>
<td><strong>Iterative Deepening</strong></td>
<td>Yes</td>
<td>Yes if uniform cost, no otherwise</td>
<td>$O(b^d)$</td>
<td>$O(bd)$</td>
</tr>
<tr>
<td><strong>Uniform Cost</strong></td>
<td>Yes iff costs are positive</td>
<td>Yes iff costs are positive</td>
<td>$O(b^{\text{max_depth}})$</td>
<td>???</td>
</tr>
</tbody>
</table>
Graph Search

- Simple modification: only add nodes to frontier if
  - they are not already in the frontier
  - and they have not already been expanded
Reading and Next Class

- Today’s material: AIMA 3–3.4
- Next class: AIMA 3.5–3.6