CS 251 | **Artificial Intelligence** | **Fall 2007**

Midterm Exam #1 | **October 15–19, 2007**

You may spend as much time working the problems as you require, but the exam is due by 5:00 p.m., Friday, October 19, 2007. *Late exams will not be accepted.*

This take-home exam is open book and open notes. Any sources used must be cited in your solutions. You many not consult any human being other than your instructor. Failure to comply with these guidelines will be viewed as a violation of academic integrity.

Each problem is equally weighted. *You must show* all *of your work to receive full credit.*

GOOD LUCK!

---

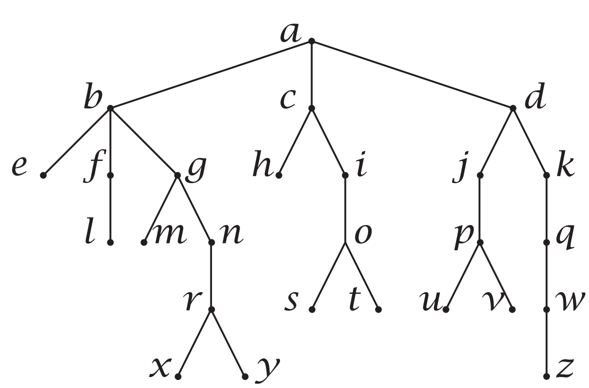
1. Show that the states of the eight-puzzle can be grouped into two disjoint subspaces, S_{odd} and S_{even}, such that no element of one subspace can be transferred into the other by any sequence of legal moves. (See problem 3.4 in Russell and Norvig.)

2. Describe the order in which nodes are expanded in the tree illustrated below for each of the following uninformed search algorithms, assuming that the left-most branch is always followed first:

   (*a*) depth-first search,

   (*b*) breadth-first search,

   (*c*) iterative deepening depth-first search (starting with an initial depth limit of 1, and incrementing the depth limit by 1 with each iteration).



3. Consider a uniform tree with branching factor b and depth D. Let $d \le D$ denote the depth of the shallowest goal node.

   (*a*) Show that the number of nodes in the tree equals $(b^{D+1} - 1)/(b - 1)$.

   For each of the following algorithms, compute upper and lower bounds for the number of nodes visited (i.e., the time complexity), and upper and lower bounds for the maximum number of nodes retained in memory at any time (i.e., the space complexity), as functions of b, d, and D:

   (*b*) depth-first search,

   (*c*) breadth-first search,

   (*d*) iterative deepening depth-first search (starting with an initial depth limit of 1, and incrementing the depth limit by 1 with each iteration).

4. Assume that the time required to discover a goal n_g at depth d by a breadth-first search for a particular problem, for which bidirectional search is not feasible, is kb^d. Consider the following hybrid algorithm: First, identify a node n at depth $\delta \approx d/2$ that is possibly on a solution path to the goal. Then, apply breadth-first search to try to find a path from n to a goal node, and then from the root node to the "sub-goal" n. (Note that both searches are conducted in a top-down direction.) If n lies on the solution path, we concatenate the two sub-paths to obtain a full solution. However, if we discover that n is not on a solution path, we revert to the original breadth-first search algorithm, starting from the root of the search tree.

   (*a*) Let c denote the time required to identify n, and let p denote the probability that n lies on a path to n_g. What conditions must c and p satisfy so that this hybrid is faster than ordinary breadth-first search on average?

   (*b*) Please provide an example problem where this new algorithm would be advantageous.

5. In Problem 4.2, on page 134, Russell and Norvig define the *heuristic path algorithm* as a best-first search algorithm with the objective function $f(n) = (2 - w)g(n) + wh(n)$, where $g(n)$ represents the path cost to reach node n from the root, $h(n)$ is a heuristic, and $w \in [0, 2]$. Assuming that h is admissible, show that the heuristic path algorithm yields an optimal path to a goal node if $0 \le w \le 1$. Conversely, show that if $w > 1$, then the algorithm is not necessarily optimal.

6. Define a nontrivial, admissible heuristic for the Towers of Hanoi puzzle with n disks and three towers.

7. Please solve problem 5.8 on page 159 of Russell and Norvig, which refers to the arc-consistency, CSP algorithm, *AC–3*, on page 146.

8. Please solve problem 5.10 on page 159 of Russell and Norvig.