CS472 Foundations of Artificial Intelligence
Fall 2001
Assignment 1 (120 pts total)

Due Wednesday, September 26 at the beginning of class.

Solutions must be typed, although equations, graphs, tables, etc., can be drawn in by hand.

1. **General AI (20 pts.)** Exercise 1.5 of Russell & Norvig. Note: Much of the needed information can be found on the WWW.

2. **General AI (10 pts.)** Exercises 1.9 and 1.10 of Russell & Norvig. Note: in your answer, contrast and compare the two questions.

3. **Uninformed search (20 pts.)** In the water-jug puzzle, we are given a 3-liter jug, named *Three*, and a 4-liter jug, named *Four*. Initially, *Three* and *Four* are empty. Either jug can be filled with water from a tap, T, and we can discard water from either jug down a drain, D. Water may also be poured from one jug into the other. There is no additional measuring device. We want to find a sequence of operations that will leave precisely two liters of water in *Four*. (First, convince yourself that this can be done!)

   a) Formulate the water-jug problem as state-space search problem. That is, define the set of states, the operators, the start state, the goal test, and the path cost.

   b) Draw a graph of all the *distinct* state-space nodes that are within three moves of the start node (state), label each node by its state description, and show at least one path to each node in the graph, by labeling each arc by the name of the appropriate operator. In addition to these nodes, draw also all of the nodes and arcs (properly labeled) on the path to the solution.

4. **Uninformed search (20 pts.)** Consider a finite tree of depth d and branching factor b, (A tree consisting of only a root node has depth zero; a tree consisting of a root node and its b successors has depth 1; etc.) Suppose the shallowest goal node is at depth $g \leq d$.

   a) What is the *minimim* and the *maximum* number of nodes that might be expanded by a depth-first search with depth bound equal to d?

   b) What is the *minimim* and the *maximum* number of nodes that might be expanded by a breadth-first search?

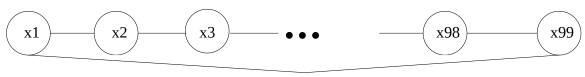
   c) What is the *minimim* and the *maximum* number of nodes that might be expanded by a depth-first iterative-deepening search?

5. **Constraint satisfaction search — problem formulation (20 pts.)** Consider the "mutilated chess board problem" (as discussed in class): A standard chess board can be tiled by 32 dominoes, each covering two squares. If two diagonally opposite squares are removed, can the remaining 62 squares be tiled by 31 dominoes?

   Formulate this problem as a Constraint Satisfaction Problem, *i.e.*, define a set variables and their domains, and a set of constraints that capture the mutilated chess board problem. Be as precise and formal as possible.

6. **Constraint satisfaction — general (10 pts.)** Exercise 3.19 of Russell & Norvig. Note: the question is about the order of variable/value assignment in general, not just for the cryptarithmetic problem.

7. **Constraint satisfaction search — backtracking (20 pts.)** Consider coloring the graph below such that no two neighboring nodes have the same color.



   a) Can this graph be colored with two colors? :-)

   b) In order to search for a two-coloring, consider a backtrack search with forward checking and vertices (nodes) assigned in order, starting at $x1$. Draw the search tree explored by the backtrack search in a search for a two-coloring. What can you say about the size of the search tree?

   c) Same question as in b), but now using a *random* ordering on the vertices to which to assign colors during the backtrack search with forward checking. (*I.e.,* at the highest level of the tree, we may assign a color to, *e.g.*, vertex $x21$ (randomly selected); at the next level down to, *e.g.*, vertex $x55$ (randomly selected); etc.) Draw an example of a search tree that one might encounter in such a backtrack search. What can you say about the size of this search tree? (No detailed calculation needed.)

   d) Same question as in c), but now add arc consistency checking at each node in the backtrack search tree.