# Homework #1: Chapters 1, 2, 3 *(revision 1)*

The following exercises are due at the beginning of class on February 1. Each exercise will be graded for correctness, so please start early and be sure you are confident in your answers. Also, remember that all work should be your own. Note this homework is continued on the reverse side. *Corrections have been made to problems #5 and #6.*

1. *[15 points]* Develop a PEAS description for the following task environments. Make sure it is clear how your performance measure would be computed. Also, state what would constitute a single percept and what would be a single action.
   a) A software agent that can play a computerized game of tic-tac-toe versus a human.
   b) A credit card fraud detection agent that examines an individual's transaction history and reports suspicious activity. The agent receives information about the individual and a complete set of transaction information from the time that the person opened a card to date. It must then decide if the transactions show a pattern of fraud.
   c) An autonomous exploration robot that seeks out signs of life on Mars.

2. *[15 points]* For each of the agents described above, categorize it with respect to the six dimensions of task environments as described on pages 41-45. Be sure that your choices accurately reflect the way you have specified your environment, especially the sensors and actuators. Give a short justification for each property.

3. *[10 points]* In what way is the table-driven agent better than the simple reflex-agent? How is the simple reflex-agent better than the table-driven agent?

4. *[10 points]* Does a finite state space always lead to a finite search tree? Explain your answer.

5. *[30 points]* Sudoku is a popular logic puzzle. Consider a 4x4 puzzle like the one given below. The object is to place the numbers 1-4 in the blank squares such that every row contains exactly one of each of the digits, and likewise for every column and each of the 2x2 quadrants. Assume that the only legal action is entering a number into the next *available* square (proceeding from left to right in each row, and moving from top to bottom). This number must not already appear in the same row, column or quadrant (which would violate the puzzle's constraints). Use breadth-first search to solve this problem. Show your search tree with each node showing the current grid and labeled with the order in which it was expanded. Hint: Your tree should have ~~8~~9 levels (not including the root node), so be sure to leave room to fit it on one sheet of paper.

Initial State *(with revisions marked)*

| | | | |
|---|---|---|---|
| 4 | | | ~~3~~ |
| | 1 | | 2 |
| ~~2~~ | 4 | 3 | |
| | | 2 | ***4*** |

Initial State *(actual)*

| | | | |
|---|---|---|---|
| 4 | | | |
| | 1 | | 2 |
| | 4 | 3 | |
| | | 2 | 4 |

6. *[20 points]* Given the map in ~~Figure 3.3 of the book (p. 63)~~ Figure 3.2 of the book (p. 68), use depth-first graph-search to find a path from Vaslui to Sibiu. Recall that in a graph-search, any state that has already been expanded will not be added to the frontier. Assume that when all else is equal, cities are chosen in alphabetical order. Show your search tree and label each node with the order in which it is expanded.