Homework #1: Chapters 1, 2, 3, and 4

The following exercises are due at the beginning of class on February 7. Each exercise will be graded for correctness, so please start early and be sure you are confident in your answers.

1. *[15 points]* Develop a PEAS description for the following task environments:
   a) An agent that can answer arbitrary trivia questions typed into a computer.
   b) A robot that can help rescue workers locate the injured in a collapsed building.
   c) A software agent that can play a computerized game of tic-tac-toe versus a human.
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 40-43. Give a short justification for each property.
3. *[15 points]* Do exercise 3.3 from the book (p. 89). Your answer can be specified either mathematically or in pseudo-code.
4. *[10 points]* Consider the problem of coloring a two-dimensional map using only four colors, such that no two adjacent areas have the same color. Give the initial state, goal test, successor function, and cost function. Specify the successor function by describing each action formally (i.e., precisely describe what kinds of states each action can be used in and how the state is changed when it is applied).
5. *[15 points]* Consider a search problem in which every state has exactly 10 successor states, none of which can be reached from any other state. Assume the maximum depth of a search is 9, and the minimum depth of a solution is 8.
   a) If we use breadth-first search, what is maximum number nodes that could be generated, and what is the maximum number of nodes that would need to be stored in memory at any given time?
   b) Answer part (a) for depth-first search.
   c) What are the comparative advantages and disadvantages of using these two search strategies for this problem?
6. *[15 points]* Use greedy search to solve the 8-puzzle with the initial and goal states shown below. Assume that your step cost is 1 per move and that your heuristic function is the number of tiles that are out of place. Show your search tree, including the h(n) value for each node, and label each node with the order in which it is expanded (note, this is different from the order it is generated). Show all generated states, including repeated states.

Initial State

1		3
4	2	5
7	8	6

Goal State

1	2	3
4	5	6
7	8	

7. *[15 points]* Consider the road map of Romania in Fig. 3.2 on p. 63 of the text book. Use A* to find a path from Dobreta to Bucharest, where step costs are as labeled on the map and the values for the heuristic function are given by Fig. 4.1 on p. 95. Show your search tree and label each node with the order in which it is expanded. Show the f(n), g(n), and h(n) values for each node. In order to reduce unnecessary search, you can ignore moves that return you to the state you just came from, however you must show any other repeated states.