Homework #1: Chapters 1, 2, 3

The following exercises are due at the beginning of class on February 5. There are exercises on both sides of this sheet. Please type your answers or neatly write them on your own paper. Each exercise will be graded for correctness, so start early and be sure you are confident in your answers. Also, remember that all work should be your own.

1. *[15 points]* Develop a PEAS description for the following task environments:

   a) A software agent that can play the game FreeCell. FreeCell (found on MS Windows machines) is a single-person card game in which you deal all cards into eight piles facing up (so that you can see every card) at the start. The object is to move all the cards to four foundations piles that are built up in suit from Ace to King. There are also four free cells that can be used as placeholders for single cards. A card can be moved to another pile if it is the next card in descending order and has the opposite color.

   b) A computer program that given an image of a fingerprint can find the best match in a database of criminal fingerprints

   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 40-43. 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. *[20 points]* Consider the following situation. A 3-foot tall monkey is in a room where some bananas are suspended from the 8-foot tall ceiling. He would like to get the bananas. The room contains two stackable, movable, climbable 3-foot-high crates. Give the initial state, goal test, successor function, and path cost function for this problem. Choose a formulation that is precise enough to be implemented. In particular, provide a mathematical description of the state and a successor function that precisely defines what kinds of states each action can be used in and how the state is changed when it is applied (you may use action/ successor pairs, mathematics and/or pseudo-code). You do not need to provide a solution (i.e., successful sequence of actions) for the problem.

5. *[25 points]* Consider the 8-puzzle with the initial and goal states shown below. Use breadth-first search to solve this problem. In order to reduce unnecessary search, you can ignore moves that return you to the state you just came from. Show your search tree, and label each node with the order in which it is expanded. To save yourself some unnecessary work, you may stop as soon as you have generated the goal state (i.e., you don't need to expand any other nodes after you have found the goal state).

Initial State

2		3
1	8	4
7	6	5

Goal State

1	2	3
8		4
7	6	5

6. *[15 points]* Suppose that LEGAL-ACTIONS(*s*) denotes the set of actions that are legal in state *s*, and RESULT(*a*, *s*) denotes the state that results from performing a legal action *a* in state *s*. Define SUCCESSOR-FN in terms of LEGAL-ACTIONS and RESULT, and *vice versa.* Your answer can be specified either mathematically or in pseudo-code.