COSC 370 Exam Review #1
Spring 2013

1.) Define the Turing Test.

2.) What do we mean by a rational agent?

3.) For the following tasks, note if they can be done at present or not (with AI) and justify:

   a. Play a decent game of table tennis
   b. Buy a week's worth of groceries at Giant
   c. Discover and prove a new mathematical theorem

4.) Give the PEAS, including a full accounting of the environment for a chess playing agent.

5.) Using pseudo-code give an example of a simple reflex agent for a 2x2 vacuum cleaner

6.) What's the difference between a goal-based agent and a utility-based agent?

7.) Give the problem formulation (goal, states, action, solution) for the problem of attempting to get from Arad, Romania to Bucharest, Romania.

8.) What are the four problem types in agent design? Define them.

9.) Give a single-state, conformant and contingency agent design for the vacuum world problem

10.) Give the state space description for the Arad to Bucharest problem.

11.) Give a tree expansion for a simple 3-puzzle where blank is in the upper left corner, 2 is in the upper right, 1 is in the bottom right and 3 is in the bottom left. What tree search technique is best in this situation? Why?

12.) Describe depth limited and iterative deepening search. Include all of the properties discussed in class.

13.) Design an A* algorithm for a robot that has to navigate from one side of a room to the other, avoiding any obstacles along the way.

14.) Would A* or Greedy search be better for the following problem: I have a weighted tree of finite depth and width and I want to find the path from a

particular vertex v that has the largest cost. Note that the algorithm would return the destination vertex in this case.

15.) What is required for a heuristic to be admissible? What is required for a heuristic to dominate another heuristic?

16.) Describe a simulated annealing solution to the classic Sudoku problem.

17.) Describe a hill-climbing solution to the n-queens problem.

18.) Describe local beam search.

19.) How do we deal with non-determinism using a search algorithm? How do we deal with actions that can occasionally fail?

20.) Without relying on pseudocode (or any other code snippets), describe the minimax approach to adversarial search.

21.) In pseudocode, describe alpha-beta pruning.

22.) What alterations would need to be made to the minimax approach if we introduce an element of chance (coin flip, die roll, etc.)?

23.) Why would we need a cutoff test for alpha-beta pruning or minimax?

24.) Give an example of a deterministic, perfect information game. Give an example of an imperfect information, deterministic game.

25.) Give an example of a constraint satisfaction problem and how it is solved.