CPSC 3750 – A.I.
Assignment 1 **Due Jan. 29**

1) **(6 pts)** Define in your own words the following terms: intelligence, artificial intelligence, rationality.

   Solution: In dictionaries, intelligence is defined as "the capacity to acquire and apply knowledge", or "the ability to comprehend and profit from experience" (which means there is a learning component involved), etc. We can also define intelligence using quantifiable terms: "the baility to use knowledge to perform better in an environment". This is also our definition of rationality.

   AI = the design of intelligent agent programs.

2) **(6 pts)** *Surely, computers cannot be intelligent. They can do only what their programmers tell them.* Do you think this statement is true? Explain your point of view.

   Solution: The answer depends on your definition of intelligence. For example, a programmer can implement a learning algorithm to solve a problem. The behaviour of this program would have little in common with the skill of that programmer in solving the problem. If you accept this *learning algorithm* as a sign of intelligence, then the answer is yes.

3) **(12 pts)** Give a PEAS description for the task environment of an internet book-shopping agent. State any assumptions that you are making about the problem. Then characterize the environment using the properties given in Section 2.3 in your textbook. What is an appropriate design for this agent? (see Section 2.4 in the text)

   Solution:

   – Environment: Internet
   – Actuators: follow links, submit data to web sites, display data to user.
   – Sensors: web page content, user requests
   – Performance measure: minimize cost, obtain books requested or deemed interesting.

   This task environment is partially observable, deterministic, discrete, and single agent. It is episodic if the goal is purchasing books at lowest price, but it is sequential if the agent learns to recommend interesting books. It is static, but you may answer *dynamic* if you consider that the stocks of online bookstores may fluctuate in real time.

   Design: utility or goal based agent, learning agent.

4) **(30 pts)** Consider the n-queens problem described on pages 66-67 in your text. Three different state spaces are suggested: (a) a complete state formulation where every state corresponds to a particular configuration of n queens on the board, (b) an incremental formulation where every state represents a board with i queens and

$0 \leq i \leq n$, and an action places a queen on an empty square. (c) an efficient incremental formulation where every state represents a board with i queens placed on columns 1 to i, $0 \leq i \leq n$, and an action places a queen on column $i+1$ so that it is not attacked by any of the other queens.

(i) For each of the three state spaces, argue whether it is possible to design: a breadth first search, an iterative deepening search, and a bidirectional search algorithm. Be brief and cover all nine combinations.

Solution: Breadth first search: a large branching factor makes this strategy impractical due to memory constraints. For complete state $b = n(n^2 - n)$ thus b is $O(n^3)$, for incremental search $b = n^2 - i$ at i-th iteration and b is $O(n^2)$, and for efficient incremental $n \geq b \geq n - 3i$ and b is $O(n)$. However, even for the efficient incremental, b is too large to make BFS feasible given a reasonable amount of memory.
Iterative deepening: for the complete state, memory requirements are not the limiting factor but time. For both incremental formulations, the state space is a tree itself, and the goal states are found at the leaves which are at depth n. iterative deepening is not more powerful than DFS.
Bidirectional search: the first requirement is to have access to the goal state. However, we do not know the goal state unless we compute it. Therefore, it is not possible to design a bidirectional search algorithm in all three state spaces.

(ii) Show that the number of states for the efficient incremental formulation is at least $O((n!)^{\frac{1}{3}})$. Estimate the largest value for n for which exhaustive exploration (enumerating all states) is feasible, i.e. you think it would complete within an hour or so on your computer. (see the hint from problem 3.5 from page 89 in the text).

Solution: At iteration i, there are $i-1$ queens on the board on columns 1 to $i-1$. Each queen attacks at most 3 squares on column i, corresponding to the row and two diagonals. Therefore there are *at least* $n - 3i$ choices for the i-th queen. To count the total number of states, we count the total number of choices to place the queens, which we denote Q_0:

$$Q_0 = n(n-3)(n-6)\ldots(n-3i)\ldots$$

We also denote,

$$Q_k = \Pi_{i=1}^{\lfloor \frac{n}{3} \rfloor}(n - 3i + k).$$

Since $n - 3(i-1) \geq n - 3i + k$ for $k \in \{1, 2\}$, it follows that $Q_0 \geq Q_1$ and $Q_0 \geq Q_2$[1]. But $n! = Q_0 Q_1 Q_2$, so $n! \leq Q_0^3$ or $Q_0 > (n!)^{\frac{1}{3}}$. More rigorously, since we are ignoring constants, we were able in fact to show that $Q_0 \geq O(n!^{\frac{1}{3}})$.

[1] If you want to be really accurate, this claim is true only up to some constants that depend on the divisibility of n by 3. For example, an accurate claim is $cQ_0 \geq Q_1$ for some constant c. But this means Q_0 is $\Omega(Q_1)$ (recall your big-oh and omega notation from your discrete math class).

(iii) For each of the three state spaces, answer the following questions. Is the state space finite or infinite? Can a search tree in the state space be finite or infinte? If it is possible to have both finite and infinite search trees in a particular state space, what conditions you need to impose on the search algorithm in order to guarantee finiteness of the search tree?

Solution: For all formulations and a fixed n, the state space is finite. A search tree in the complete formulation can be infinite. The states on the search tree will be repeated. If we implement the graph search algorithm described in the text (no repeateed states), then the search tree becomes finite as well.
For both incremental formulations, the branching factor is finite and the depth of any search tree is at most n, so the search trees are finite.

5) **(10 pts)** In bi-directional search, two searches occur in parallel, one from the initial state, the other from the goal state. Which of the following search algorithms can be used for the parallel searches and why: breadth first search, depth first search, depth limited search, iterative deepening depth first search.

Solution: It is possible to use breadth first search (see text). Depth limited search has the disadvantage of not guaranteeing to find a solution, so it cannot be used. Iterative deepening is a DFS simulation of BFS, so it can be used, provided that the searches occur in parallel based on the depth limit. DFS cannot be used because search depth needs to be controlled and this is not the case with pure DFS.

6) **(26 pts)** Consider the vacuum cleaner world described in your text on page 33 with one modification. The performance measure assigns +1 for each clean room in every time unit and -1 for every move of the agent.

(i) Can a simple reflex agent be rational? Explain.

Solution: A simple reflex agent cannot avoid oscillating between the rooms.

(ii) What about a model based reflex agent, using states? If the answer is yes, design such an agent, explain your design and argue why the agent is rational.

Solution: Yes, one can use the states to remember that a room was visited and it is now clean.

(iii) How do your previous answers change (if they indeed change) if the agent can perceive the status of both rooms instead of only the status of the room the agent is in.

Solution: A simple reflex agent can then be rational. The answer doesn't change for the model based reflex agent.