Artificial Intelligence CSE 537
Assignment 1 (Sept. 24th 2015)

Due date and time: Oct. 12th in class.
Submit in class (hardcopy).
* 80 points max

Question 1: Warm-up (1 point)
What is your preferred e-mail contact?

Question 2: Intelligent Agents (9 points)
Develop a PEAS description of the task environment and characterize the type of environment along the seven dimensions: 1) fully vs. partially observable, 2) deterministic vs. stochastic, 3) episodic vs. sequential, 4) static vs. dynamic, 5) discrete vs. continuous 6) benign vs. adversarial and 7) known vs. unknown. Provide explanation for picking the six dimensions.

(a)[3 points] Exploring the subsurface oceans of Titan
(b)[3 points] Traveling salesman problem
(c)[3 points] Practicing tennis against a wall

* TSP problem is finding the minimal distance tour starting and finishing at a specified vertex after having visited each other vertex exactly once. Assume that it is represented as an undirected weighted graph, such that cities are the graph's vertices, paths are the graph's edges, and a path's distance is the edge's length.

Question 3: Search (20 points – two points for each “_____”)
Consider the search space below, where S is the start node and G1, G2, and G3 satisfy the goal test. Edges are labeled with the cost of traversing them and the estimated cost to a goal is reported inside nodes.
For each of the following search strategies, indicate which goal state is reached (if any) and list, in order, all the states popped off of the *frontier* list.

<table>
<thead>
<tr>
<th>Search Strategy</th>
<th>Goal state reached:</th>
<th>States popped off <em>frontier</em>:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Uniform Cost</td>
<td>_______</td>
<td>___________________________</td>
</tr>
<tr>
<td>Iterative Deepening</td>
<td>_______</td>
<td>___________________________</td>
</tr>
<tr>
<td>Greedy Best First</td>
<td>_______</td>
<td>___________________________</td>
</tr>
<tr>
<td>A*</td>
<td>_______</td>
<td>___________________________</td>
</tr>
<tr>
<td>Hill Climbing</td>
<td>_______</td>
<td>___________________________</td>
</tr>
</tbody>
</table>

**Question 4: Search (5 points)**

Draw and explain a simple state space that shows why in general we need to check for goal states when a node is removed from *frontier* rather than when it is added. (Do not include more than five nodes in your state space.)

**Question 5: Heuristic Functions (5 points)**

Assume $h_1$ and $h_2$ are admissible heuristic functions. Let $h_3 = \max(h_1, h_2)$ and $h_4 = \min(h_1, h_2)$. Is $h_3$ admissible? Is $h_4$? Which of the following would likely work best: $h_1$, $h_3$, or $h_4$? Explain your answers. * All these $h$’s take a node as an argument.

**Question 6: Simulated annealing (5 points)**

Consider simulating annealing (SA) algorithm for a Traveling Salesman Problem (TSP). Describe how you would define the following components in simulate annealing algorithm for solving the TSP. Provide explanations.

(a)[1 point] Represent the tour (state and data structure of the state)
(b)[1 point] Generate random rearrangement of the tour
(c)[1 point] Energy function (objective function)
(d)[1 point] Goal test
(e)[2 points] rewrite the SA pseudo-code using your definitions

**Question 7: Genetic Algorithm (5 points)**

Consider simulating annealing algorithm for a Traveling Salesman Problem (TSP). Describe how you would define the following components in Genetic algorithm for solving the TSP. Provide explanations.

Fitness function (objective function)
(a)[1 point] Represent the tour (state and data structure of the state)
(b)[1 point] Cross-over
(c)[1 point] Mutation (including how you would define the random probability)
(d)[1 point] Fitness function
(e)[1 point] Goal test
(f)[2 points] rewrite the SA pseudo-code using your definitions
Question 8: Minimax Algorithm (15 points)
Consider the two-player game in the following figure

Figure 1. The starting position of a simple game. Player A moves first. The two players take turns moving, and each player must move his token to an open adjacent space in either direction. If the opponent occupies an adjacent space, then a player may jump over the opponent to the next open space if any. (For example, if A is on 3 and B is on 2, then A may move back to 1.) The game ends when one player reaches the opposite end of the board. If player A reaches space 4 first, then the value of the game to A is +1; if player B reaches space 1 first, then the value of the game to A is −1.

A. Draw the complete game tree, using the following conventions:
   - Write each state as (sA,sB), where sA and sB denote the token locations.
   - Put each terminal state in a square box and write its game value in a circle.
   - Put loop states (states that already appear on the path to the root) in double square boxes. Since their value is unclear, annotate each with a “?” in a circle.

B. Now mark each node with its backed-up minimax value (also in a circle). Explain how you handled the “?” values and why.

C. Explain why the standard minimax algorithm would fail on this game tree and briefly sketch how you might fix it, drawing on your answer to (b). Does your modified algorithm give optimal decisions for all games with loops?

Question 8: Constraint Satisfaction Problems (15 points)
Consider the graph with 8 nodes A₁, A₂, A₃, A₄, H, T, F₁, F₂. A₁ is connected to Aᵢ₊₁ for all i, each Aᵢ is connected to H, H is connected to T, and T is connected to each Fᵢ. Find a 3-coloring of this graph by hand using the following strategy: backtracking with conflict-directed backjumping, the variable order A₁, H, A₄, F₁, A₂, F₂, A₃, T, and the value order R, G, B.