CS 331 Midterm
Spring 2011

You have 50 minutes to complete this midterm. You are only allowed to use your textbook, your notes, your assignments and solutions to those assignments during this midterm. If you find that you are spending a large amount of time on a difficult question, skip it and return to it when you’ve finished some of the easier questions. Total marks for this midterm is 50.

<table>
<thead>
<tr>
<th>Section</th>
<th>Marks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agents</td>
<td>/ 14</td>
</tr>
<tr>
<td>Search</td>
<td>/ 22</td>
</tr>
<tr>
<td>Games</td>
<td>/ 14</td>
</tr>
<tr>
<td>Total</td>
<td>/ 50</td>
</tr>
</tbody>
</table>
Section I: Agents (14 points)
1. In this question, you will answer questions about an agent that is operating in the Tragedy of the Commons environment exactly as we described it in class. For each part below, circle the choice which best describes the environment for an agent operating in the Tragedy of the Commons assuming only one “game” is played. Write a one or two sentence justification.

a) Fully observable or Partially observable [2 points]

b) Deterministic or Stochastic [2 points]

c) Episodic or Sequential [2 points]

d) Static or Dynamic [2 points]

e) Is the strategy space for a player (as described in class) Discrete or Continuous [2 points]

f) Single agent or Multi-agent [2 points]

2. Suppose you had a checkers playing agent. When would you prefer a utility-based agent over a goal-based agent? [2 points]
Section II: Search [22 points]
1. A “greedy” algorithm is a “short-sighted” algorithm that always picks the best option out of a set of possibilities. This set of possibilities is “short-sighted” because it does not consider all possibilities but rather a subset of them for computational efficiency. Which of the following are “greedy”? You may circle more than one. [4 points]

- Hillclimbing
- Simulated Annealing
- Breadth-First Search
- Gradient Descent

2. What is the main drawback to a greedy algorithm? [2 points]

3. The evaluation function $f(n)$ for A-star consists of two parts as shown below:

$$f(n) = g(n) + h(n)$$

In the function above, $g(n)$ is the cost to get to node $n$ from the initial state and $h(n)$ is the heuristic value containing an estimate of how far away $n$ is to the goal.

a) If we set $g(n) = 0$ for all nodes $n$, what is the name of the search algorithm? Does this algorithm find the optimal solution? [2 points]

b) If we set $h(n) = 0$ for all nodes $n$, and if the step-cost is always 1 to go from one state to its immediate successor, what algorithm do we end up with? Does this algorithm find the optimal solution? [2 points]

c) Suppose we use A-star to solve the 3x3 tile puzzle. If we set $h(n) = 1000$ for all nodes $n$, does this algorithm find an optimal solution? Explain your answer. [2 points]
4. In class, we pointed out that the time complexity of both Iterative Deepening DFS and A-Star search was $O(b^d)$, where $b =$ the branching factor and $d =$ the depth of the shallowest goal node. However, if you have a good heuristic, A-Star search is often much faster than ID-DFS even though their time complexity is the same. How do you explain this discrepancy? [2 points]

5. Suppose you want to find an optimum point for the function $f(x,y) = 3x^3y + y^2 + 1$. If you initialize gradient descent to start at (1,1), and you run gradient descent for 1 iteration, what $(x,y)$ coordinates do you end up at? Use alpha = 0.1. [8 points]
Section III: Games [14 points]

1. Circle all the pure strategy Nash equilibria in the matrix below. If there are none, write “No Nash Equilibrium”. [6 points]

<table>
<thead>
<tr>
<th></th>
<th>B: S1</th>
<th>B: S2</th>
<th>B: S3</th>
</tr>
</thead>
<tbody>
<tr>
<td>A: S1</td>
<td>A = 7, B = -7</td>
<td>A = -10, B = -10</td>
<td>A = 5, B = 3</td>
</tr>
<tr>
<td>A: S2</td>
<td>A = -6, B = 6</td>
<td>A = 2, B = -2</td>
<td>A = -2, B = 2</td>
</tr>
<tr>
<td>A: S3</td>
<td>A = 100, B = 1</td>
<td>A = 0, B = 0</td>
<td>A = 1, B = -3</td>
</tr>
</tbody>
</table>

2. Take a look at the game tree below. The rectangular nodes correspond to terminal nodes with their utility values.
   a) What is the minimax value of the root? Fill in the minimax values for each circular node. [3 points]
   b) If you use alpha-beta pruning, which branches can be pruned? [5 points]