1. NAME: _____________________________________________________________

YOUR ID: ___________ ID TO RIGHT: ___________ ROW: ______ NO. FROM RIGHT: ______

2. (15 pts total, 1 pt each) Label the following task environment properties as shown.

<table>
<thead>
<tr>
<th>Task Environment</th>
<th>Observable Fully(F)</th>
<th>Deterministic Deterministic(D)</th>
<th>Episodic Episodic(E)</th>
<th>Static Static(Stat)</th>
<th>Discrete Discrete(D)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Crossword Puzzle</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Taxi Driving</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Part-picking Robot</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

3. (10 pts total) In general, which is the preferred uninformed search method when (1) there is a large search space, (2) the depth of the solution is unknown, and (3) an optimal solution is unnecessary (i.e., any solution will do)? (Mark one blank with “X”)

_____ Depth-first search   _____ Breadth-first search   _____ Uniform-cost search

_____ Depth-limited search   _____ Iterative-deepening search

4. (8 pts total, 2 pts each) Your book defines a task environment as a set of four things, with the acronym PEAS. Fill in the blanks with the names of the PEAS components.

P____________________  E____________________  A____________________  S____________________

5. (32 pts total, 2 pts each) Fill in the values of the four evaluation criteria for each search strategy shown. Assume a tree search where b is the finite branching factor; d is the depth to the shallowest goal node; m is the maximum depth of the search tree; l is the depth limit; step costs are identical and equal to some positive ε; and in Bidirectional search both directions use breadth-first search.

Note: These conditions are the same as in Figure 3.21 of your textbook.

<table>
<thead>
<tr>
<th>Criterion</th>
<th>Complete?</th>
<th>Time complexity</th>
<th>Space complexity</th>
<th>Optimal?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Uniform-Cost</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Depth-Limited</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Iterative Deepening</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bidirectional (if applicable)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**** TURN PAGE OVER AND CONTINUE ON THE OTHER SIDE ****
6. (35 pts total, -5 for each wrong answer, but not negative)
Simulate A* on the following graph to find the optimal path from the start state, S, to one of the
goal states, G1 and G2. Each state is labeled X/N where X is the name of the state and N is a
heuristic estimate of the remaining distance to the closest goal state. Arrows lead from a state to
its successors. Each arrow is labeled with its step cost.
At each iteration, indicate (1) the node popped off the queue, (2) its children, and (3) the
resulting queue. Show each node as (X/f/g/h) where X is the name of the state, f is the estimated
total path cost, g is the path cost so far, and h is the heuristic estimate of the remaining distance to
the closest goal state. You may not need all steps shown.
The first iteration is done for you as an example.

Initial Queue: (S/16/0/16)

Popped Node: (S/16/0/16)

Children (order doesn't matter): (B/19/5/14) (A/17/3/14)

Queue (order matters): (A/17/3/14) (B/19/5/14)

Popped Node: ________________________________

Children: ________________________________

Queue: ________________________________

Popped Node: ________________________________

Children: ________________________________

Queue: ________________________________

Popped Node: ________________________________

Children: ________________________________

Queue: ________________________________

Popped Node: ________________________________

Children: ________________________________

Queue: ________________________________

Popped Node: ________________________________

Children: ________________________________

Queue: ________________________________