1. NAME AND EMAIL ADDRESS: ___________________________________________________ 

YOUR ID: _______ ID TO RIGHT:_______ ROW:____ NO. FROM RIGHT:____ 

2. (30 pts total, 2 pts each) Label the following task environment properties as shown. The first one is done for you as an example. (From Fig. 2.6, p. 45, in R&N.)

<table>
<thead>
<tr>
<th>Task Environment</th>
<th>Observable</th>
<th>Deterministic</th>
<th>Episodic</th>
<th>Static</th>
<th>Discrete</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Fully (F)</td>
<td>Deterministic (D)</td>
<td>Episodic (E)</td>
<td>Static (Stat)</td>
<td>Discrete (Continuous)</td>
</tr>
<tr>
<td>Crossword Puzzle</td>
<td>F</td>
<td>D</td>
<td>S</td>
<td>Stat</td>
<td>D</td>
</tr>
<tr>
<td>Taxi Driving</td>
<td>P</td>
<td>S</td>
<td>S</td>
<td>D</td>
<td>C</td>
</tr>
<tr>
<td>Chess with a clock</td>
<td>F</td>
<td>D</td>
<td>S</td>
<td>Semi [D OK]</td>
<td>D [C OK]</td>
</tr>
<tr>
<td>Part-picking Robot</td>
<td>P</td>
<td>S</td>
<td>E</td>
<td>D</td>
<td>C</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 _____ X _____ Iterative-deepening search

(See the last line of the penultimate paragraph of Sec. 3.4.5, p. 90, in R&N.)

4. (30 pts total, -5 each wrong answer, but not negative) Assume that (1) the state space is infinitely deep, (2) the branching factor is finite, (3) there are cycles and loops, (4) multiple goal nodes exist at different depths with different costs, and (5) each step cost is a variable positive number \( \geq \varepsilon > 0 \). (“Y” = yes, “N” = no)

(R&N Fig.3.21, p. 91; footnote ‘c’ fails so BFS & Iterative-deepening not optimal.)

4a. Is depth-first search complete? _____ N _____ optimal? _____ N _____.

4b. Is breadth-first search complete? _____ Y _____ optimal? _____ N _____.

4c. Is uniform-cost search complete? _____ Y _____ optimal? _____ Y _____.

4d. Is depth-limited search complete? _____ N _____ optimal? _____ N _____.

4e. Is iterated-deepening search complete? _____ Y _____ optimal? _____ N _____.

**** TURN PAGE OVER AND CONTINUE ON THE OTHER SIDE ****
5. (30 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. (See Sec. 3.5.2, p. 93, R&N.)

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: (A/17/3/14)

Children: (C/20/10/10)

Queue: (B/19/5/14) (C/20/10/10)

Popped Node: (B/19/5/14)

Children: (C/35/25/10)

Queue: (C/20/10/10) (C/35/25/10)

Popped Node: (C/20/10/10)

Children: (G1/27/27/0) (G2/24/24/0)

Queue: (G2/24/24/0) (G1/27/27/0) (C/35/25/10)

Popped Node: (G2/24/24/0) => Done, the search stops here.

Children: 

Queue: (Not necessary to show the final queue, but OK if you did) 