CS-171, Intro to A.I., Winter Quarter, 2012 — Quiz # 1 — 20 minutes

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.
(From Fig. 2.6, p. 45, in R&N.)

Task Environment	**Observable** Fully(**F**) Partially(**P**)	**Deterministic** Deterministic(**D**) Stochastic(**S**)	**Episodic** Episodic(**E**) Sequential(**S**)	**Static** Static(**Stat**) Semi(**Semi**) Dynamic(**D**)	**Discrete** Discrete(**D**) Continuous(**C**)
Crossword Puzzle	F	D	S	Stat	D
Taxi Driving	P	S	S	D	C
Part-picking Robot	P	S	E	D	C

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. (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.

Performance (measure) Environment Actuators Sensors
(See p. 40 in R&N)

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.

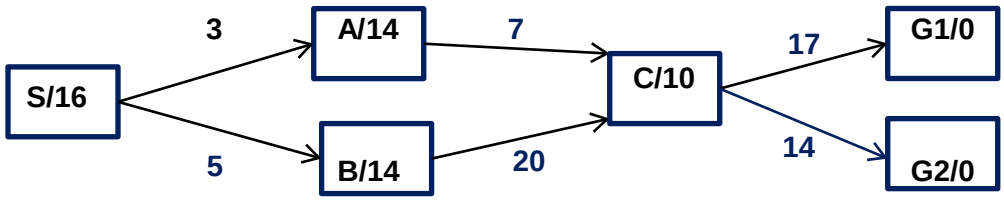
Criterion	Complete?	Time complexity	Space complexity	Optimal?
Uniform-Cost	Yes	O(b^(1+floor(C*/ε))) O(b^(d+1)) also OK	O(b^(1+floor(C*/ε))) O(b^(d+1)) also OK	Yes
Depth-Limited	No	O(b^l)	O(bl)	No
Iterative Deepening	Yes	O(b^d)	O(bd)	Yes
Bidirectional (if applicable)	Yes	O(b^(d/2))	O(b^(d/2))	Yes

****** 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. **(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: (None) .

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

Popped Node: .

Children: .

Queue : .