CS-171, Intro to A.I., Fall Quarter, 2013 — Quiz # 1 — 20 minutes

1. NAME:_____

YOUR ID: _____ ID TO RIGHT:_____ ROW:_____ NO. FROM RIGHT:_____

2. (30 pts total, 2 pts each) For each of the following terms on the left, write in the letter corresponding to the best answer or the correct definition on the right. The first one is done for you as an example.

A	Agent	A	Perceives environment by sensors, acts by actuators
K	Percept	B	All states reachable from the initial state by a sequence of actions
P	Performance Measure	C	Guaranteed to find a solution if one is accessible
L	Rational Agent	D	Process of removing detail from a representation
B	State Space	E	Maximum number of successors of any node
I	Search Node	F	Set of all pending nodes available for expansion at any given time
N	Link between nodes	G	Estimates cost of cheapest path from current state to goal state
J	Path	H	Guaranteed to find lowest cost among all accessible solutions
D	Abstraction	I	Represents a state in the state space
H	Optimal Search	J	Sequence of states connected by a sequence of actions
C	Complete Search	K	Agent's perceptual inputs at any given instant
M	Expand a state	L	Agent that acts to maximize its expected performance measure
F	Frontier	M	Apply each legal action to a state, generating a new set of states
O	Search Strategy	N	Represents an action in the state space
E	Branching Factor	O	How a search algorithm chooses which node to expand next
G	Heuristic Function	P	Evaluates any given sequence of environment states for utility

3. (5 pts each, 25 pts total) Recall that

- True path cost so far to node n = $g(n)$.
- Estimated optimal cost to goal from node n = $h(n)$.
- Estimated total cost of optimal path through node n = $f(n) = g(n) + h(n)$.

The following is a proof that A* tree search (queue sorted by f(n)) is optimal if the heuristic is admissible. The lines of the proof have been labeled A through G. Unfortunately, the lines have been scrambled. Let *ng* be the first goal node popped off the queue. Let *no* be any other node on the queue. We wish to prove that *no* can never be extended to a path to any goal node that costs less than the path to *ng* that we just found.

A : true total cost of optimal path to *ng*
F: = $g(ng)$ // because *ng* represents a complete path
D: = $f(ng)$ // by definition of *f(ng)* with $h(ng) = 0$ because *ng* is a goal node
B : ≤ $f(no)$ // because queue is sorted by *f()*
E: = $g(no) + h(no)$ // by definition of *f()*
C : ≤ $g(no)$ + true cost to goal from *no* // because *h()* is admissible
G : = true total cost of *no*

Fill in the blanks with the letters B, C, D, E, and F to prove that the true total cost of *ng* ≤ true total cost of *no*. The first and last letters, A and G, are done for you as an example.

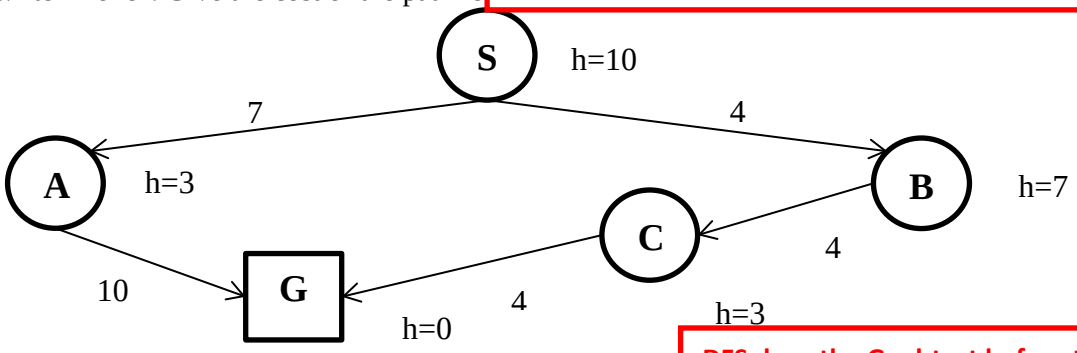
A **F** **D** **B** **E** **C** **G**

****** TURN PAGE OVER AND CONTINUE ON THE OTHER SIDE ******

5. (45 pts total, 9 pts each) Execute Tree Search through this graph (i.e., do not remember visited nodes). Step costs are given next to each arc. Heuristic values are given next to each node (as h=x). The successors of each node are indicated by the arrows out of tha

For each search strategy below, show the
means that its children are generated), ending with
goal, or write "None". Give the cost of the path fo

Please see the lecture slides for Uninformed Search, topic "When to do Goal-Test? When generated? When popped?" for clarification about exactly what to do in practical cases.



5.a. DEPTH FIRST SEARCH.

See Section 3.4.3 and Fig. 3.17.

DFS does the Goal-test before the child is pushed onto the queue. The goal is found when A is expanded.

Order of node expansion: S A G

Path found: S A G Cost of path found: 17

5.b. (9 pts) BREADTH FIRST SEARCH.

See Section 3.4.1 and Fig. 3.11.

BFS does the Goal-test before the child is pushed onto the queue. The goal is found when A is expanded.

Order of node expansion: S A G

Path found: S A G Cost of path found: 17

5.b. (9 pts) UNIFORM COST SEARCH.

See Section 3.4.2 and Fig. 3.14.

UCS does goaltest when node is popped off queue.

Order of node expansion: S B A C G

Path found: S B C G Cost of path found: 12

5.c. (9 pts) GREEDY (BEST-FIRST) SEAR

GBFS has the same behavior whether the goaltest is done before node is pushed or after node is popped, because h=0 for a goal node, so goal nodes always sort to the front of the queue anyway.

Order of node expansion: S A G

See Section 3.5.1 and Fig. 3.23.

Path found: S A G Cost of path found: 17

5.d. (9 pts) ITERATED DEEPENING SEARCH.

IDS does the Goal-test before the child is pushed onto the queue. The goal is found when A is expanded.

Order of node expansion: S S A G

See Sections 3.4.4-5 and Figs. 3.18-19.

Path found: S A G Cost of path found: 17

5.e. (9 pts) A* SEARCH.

A* does goaltest when node is popped off queue.

Order of node expansion: S A B C G

See Section 3.5.2 and Figs. 3.24-25.

Path found: S B C G Cost of path found: 12

Is the heuristic admissible (Yes or No)? Yes