CS-171, Intro to A.I., Spring Quarter, 2011 — Quiz # 1 — 20 minutes

1. (2 pts) NAME AND EMAIL ADDRESS: ______________________________

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

Performance measure Environment Actuators Sensors

4. (30 pts total, -2 for each wrong answer, but not negative) 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.

Criterion	~~Complete?~~	Time complexity	Space complexity	~~Optimal?~~
Breadth-First	~~Yes [a]~~	$O(b^d)$	$O(b^d)$	~~Yes [c]~~
Uniform-Cost	~~Yes [a,b]~~	O(b^(1+floor(C*/ε)))	O(b^(1+floor(C*/ε)))	~~Yes~~
Depth-First	~~No~~	O(b^m)	O(bm)	~~No~~
Iterative Deepening	~~Yes [a]~~	O(b^d)	O(bd)	~~Yes [c]~~
Bidirectional (if applicable)	~~Yes [a,d]~~	O(b^(d/2))	O(b^(d/2))	~~Yes [c,d]~~

[a] if b is finite

[b] if step costs ≥

[c] if step costs a

[d] if both directi

Complete? and Optimal? were discarded due to confusion about the footnotes.

Please note that the footnotes were not really needed, anyway, because the problem statement above already contained all of their stated conditions.

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

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. (30 pts total, -2 for each wrong answer, but not negative) Use the following tree to indicate the order that nodes are expanded, for different types of search. Assume that A is the start node and G (double box) is the only goal node. Here, path costs are shown to the right of each path, g = cost of path so far, h = estimate of remaining cost to goal, f = estimate of total path cost.

a. Uniform-cost search.

See section 3.4.2.

A, C, F, G

UCS does goaltest when node is popped off queue.

b. Iterative deepening depth-first search.

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

A; A, B, C, G

See section 3.4.5

c. Greedy best-first search.

See section 3.5.1.

A, B, E, D, C, G

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.

d. A* search.

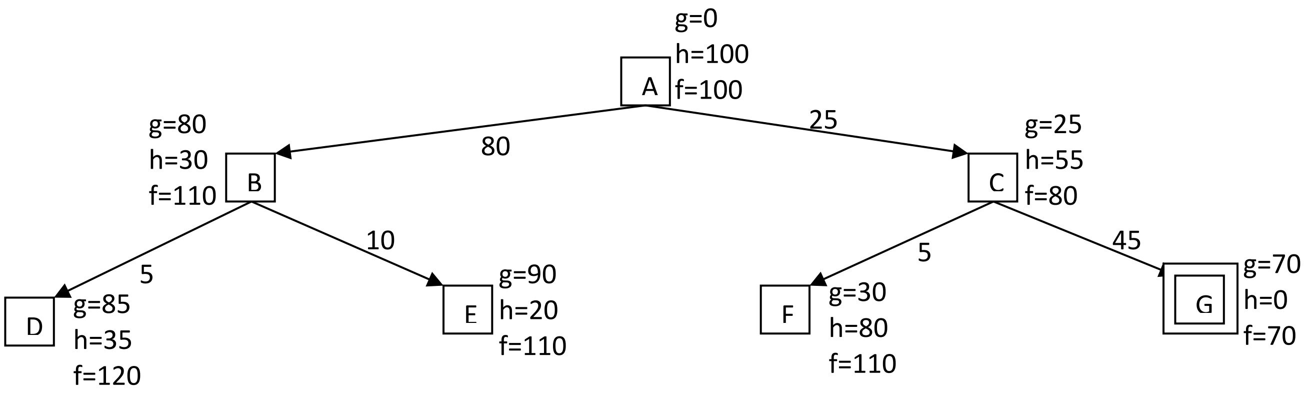
See section 3.5.2.

A, C, G

A* does goaltest when node is popped off queue.

e. Is the heuristic h admissible? (Y or N) __N__

For example, h(A)=100, but the optimal cost to the goal node G is only 70. Thus, h(A) sometimes OVER-ESTIMATES the remaining optimal distance to G, and so it is not admissible.



g=0
h=100
A
f=100
80
25
g=80
h=30
B
f=110
g=25
h=55
C
f=80
10
5
5
45
D
g=85
h=35
f=120
E
g=90
h=20
f=110
F
g=30
h=80
f=110
G
g=70
h=0
f=70