For each question on Quiz #1, “Zero” gives the percentage of students who received zero, “Partial” gives the percentage who received partial credit, and “Perfect” gives the percentage who received 100%. (Due to rounding, numbers shown below are only an approximate estimate.)

Problem 1
Zero: ~5% (~4 students), Partial: ~58% (~48 students), Perfect: ~37% (~30 students)

Problem 2
Zero: ~6% (~5 students), Partial: ~7% (~6 students), Perfect: ~87% (~71 students)

Problem 3
Zero: ~3% (~2 students), Partial: ~96% (~79 students), Perfect: ~1% (~1 student)

CS-171, Intro to A.I., Winter Quarter, 2015 — Quiz # 1 — 20 minutes

NAME: _____

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

1. (24 pts total, 1 pt each) Search Properties. Fill in the values of the four evaluation criteria for each search strategy shown. Assume a t… branching factor; d is the depth to the shallowest go… the search tree and may be infinite; l is the depth lim… to some positive ε; in bidirectional search both direc…

Your answer will be considered correct if it differs from that shown below by no more than ±1, e.g., O(b^d) vs. O(b^(d+1)).

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

	Complete?	Time complexity	Space complexity	Optimal?
Depth-First	No	O(b^m)	O(bm)	No
Breadth-First	Yes	O(b^d)	O(b^d)	Yes
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

2. (16 pts total, 4 pts each) Task Environment. 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

After more careful review and consideration, "Performance" will always receive full credit. See p. 40, section 2.3.1, where the two terms are used interchangeably. Thanks to bright, clever, and attentive students for helping to improve the class material in fine detail.

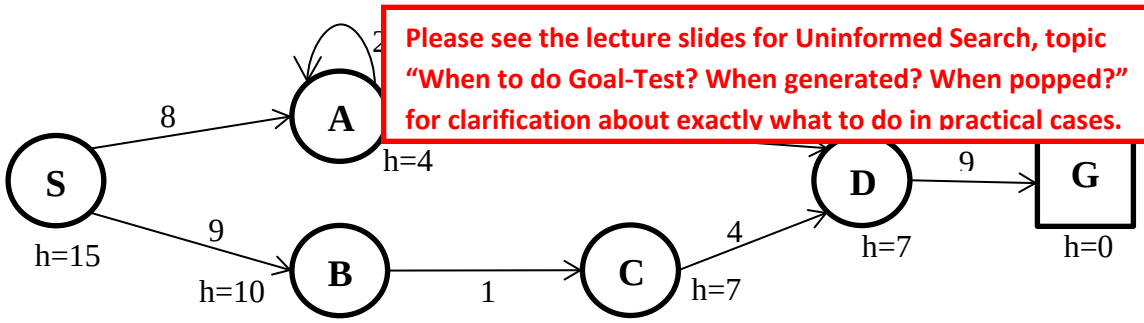
AND CONTINUE ON THE OTHER SIDE **

3. (60 pts total, 10 pts each) Execute Tree Search through this graph (i.e., do not remember visited nodes). S… …ic values are … …of that node. S… …the children … … the order in … …means that its children are generated), ending with the goal node that is found. Show the path from start to goal, or write "None". Give the cost of the path found, if any. The first one is done for you, as an example.

Minor errors will receive partial credit. The TA will determine how much credit to allow in any particular case.

It is OK if you wrote SAAAA... instead of None for "Path found." It is OK if you said N/A for "Cost of path found," or left it blank.

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.



3.a. DEPTH FIRST SEARCH.

Order of node expansion: S A A A A ...

Path found: None Cost of path found: None

3.b. (10 pts) BREADTH FIRST SEARCH.

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

Order of node expansion: S A B A D G

Path found: S A D G Cost of path found: 37

Due to discrepancies with the previous tests supplied as study guides, for Quiz #1 only, S A B A D C A D G will receive full credit.

3.b. (10 pts) UNIFORM COST SEARCH.

UCS does the Goal-test when the node is popped off the queue.

…node expansion: S A B C D G

…d: S B C D G Cost of path found: 23

3.c. (10 pts) GREEDY (BEST-FIRST) SEARCH.

Order of node expansion: S A A A A ...

Path found: None Cost of path found: None

3.d. (10 pts) ITERATED DEEPENING SEARCH

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

Order of node expansion: S S A B S A A D G

Path found: S A D G Cost of path found: 37

Due to discrepancies with the previous tests supplied as study guides, for Quiz #1 only, S S A B S A A D B C S A A A D D G will receive full credit.

3.e. (10 pts) A* SEARCH.

A* does the Goal-test when the node is popped off the queue.

…node expansion: S A B C D G

…d: S B C D G Cost of path found: 23

3.f. (10 pts total, 5 pts each) Is the heuristic admissible at nodes B & C? (Yes or No) Yes

h(B) = 10 < h*(B) = 14
h(C) = 7 < h*(C) = 13

Is the heuristic consistent at nodes B & C? No

h(B) = 10 > c(B,C) + h(C) = 1 + 7 = 8

A question arose about IDS search (question 3.d above). For clarity:

(1) Please review Fig. 3.17 & Fig. 3.18 in your textbook.

(2) Please remember that IDS begins with L=0, not L=1. This is so that you can solve trick problems like, "Starting at Arad, go to Arad." At L=0 the start node is tested to see if it is a goal, but it is not expanded (= no children are generated).

(3) Please follow along Fig. 3.17 in your textbook as we work the question in detail. (Note that I ignore their cutoff and cutoff_occurred variables for simplicity because they do not play a role below anyway.)

(3.0) Do Recursive-DLS [called RDLS below] on start node S with limit=0. Goal-test on S fails. limit=0 so return. No nodes were expanded (= no children were generated).
Nodes expanded this iteration=NIL.
Cumulative order of node expansion=NIL

(3.1) Do RDLS on S with limit=1. Goal-test on S fails. limit=1 so continue.
Expand S to yield children A, B.
Do RDLS on A with limit=0. Goal-test on A fails. limit=0 so return. (Do not expand A, i.e., do not generate A's children.)
Do RDLS on B with limit=0. Goal-test on B fails. limit=0 so return. (Do not expand B, i.e., do not generate B's children.)
Nodes expanded this iteration=S.
Cumulative order of node expansion=S.

(3.2) Do RDLS on S with limit=2. Goal-test on S fails. limit=2 so continue.
Expand S to yield children A, B.
(3.2.1) Do RDLS on A with limit=1. Goal-test on A fails. limit=1 so continue.
Expand A to yield children A, D.
Do RDLS on A with limit=0. Goal-test on A fails. limit=0 so return. (Do not expand A, i.e., do not generate A's children.)
Do RDLS on D with limit=0. Goal-test on D fails. limit=0 so return. (Do not expand D, i.e., do not generate D's children.)
(3.2.2) Do RDLS on B with limit=1. Goal-test on B fails. limit=1 so continue.
Expand B to yield child C.
Do RDLS on C with limit=0. Goal-test on C fails. limit=0 so return. (Do not expand C, i.e., do not generate C's children.)
Nodes expanded this iteration=SAB.
Cumulative order of node expansion=SSAB.

(3.3) Do RDLS on S with limit=3. Goal-test on S fails. limit=3 so continue.
Expand S to yield children A, B.
(3.3.1) Do RDLS on A with limit=2. Goal-test on A fails. limit=2 so continue.
Expand A to yield children A, D.
(3.3.1.1) Do RDLS on A with limit=1. Goal-test on A fails. limit=1 so continue.
Expand A to yield children A, D.
Do RDLS on A with limit=0. Goal-test on A fails. limit=0 so return. (Do not expand A, i.e., do not generate A's children.)
Do RDLS on D with limit=0. Goal-test on D fails. limit=0 so return. (Do not expand D, i.e., do not generate D's children.)
(3.3.1.2) Expand D to yield child G.
Do RDLS on G with limit=0. Goal-test on G succeeds. Return G as the search goal result that was found.
Nodes expanded this iteration=SAADG.
Cumulative order of node expansion=SSABSAADG.