Keerthi Teja Konuri

H Weekly Activity & Quiz Week04 Activity 9/19 **Review Test Submission: Week04 Quiz Ch03 (1)**

Review Test Submission: Week04 Quiz Ch03 (1)

User	Keerthi Teja Konuri
Course	CS 6364.001 - Artificial Intelligence - F15
Test	Week04 Quiz Ch03 (1)
Started	9/19/15 2:26 PM
Submitted	9/19/15 2:44 PM
Due Date	9/19/15 11:59 PM
Status	Completed
Attempt Score	31 out of 33 points
Time Elapsed	18 minutes out of 40 minutes
Results Displayed	All Answers, Submitted Answers, Correct Answers

Question 1

10 out of 10 points

Select the best choice for each question.

Question	Correct Match	Selected Match
Breadth-first search is complete	✓ B. Yes	✓ B. Yes
Uniform-cost search is complete	✓ B. Yes	✓ B. Yes
Depth-first search is complete	✓ A. No	✓ A. No
Depth-limited search is complete	✓ A. No	✓ A. No
Iterative-deepening search is complete	✓ B. Yes	✓ B. Yes
Breadth-first search is optimal	✓ B. Yes	✓ B. Yes
Uniform-cost search is optimal	✓ B. Yes	✓ B. Yes
Depth-first search is optimal	✓ A. No	✓ A. No
Depth-limited search is optimal	✓ A. No	✓ A. No
Iterative-deepening search is optimal	✓ B. Yes	✓ B. Yes

All Answer Choices

A. No

B. Yes

Question 2

13 out of 13 points

The functions for a node n are:
g(n) - the cost to reach the node (from the start)
h(n) - the cost to get from the node to the goal (often an estimate).
f(n) - an evaluation function which is the sum of g(n) and h(n)

Select the best choice for each informed search strategy.

Question	Correct Match	Selected Match
___ tries to expand the node that is closest to the goal, on the grounds that this is likely to lead to a solution quickly. Thus, it evaluates nodes by using just the heuristic function; that is, f (n) = h(n).	✓ D. Greedy best-first search	✓ D. Greedy best-first search
____ evaluates nodes by combining g(n), the cost to reach the node, and h(n), the cost to get from the node to the goal: f(n) = g(n) + h(n).	✓ H. A* search	✓ H. A* search
___ is identical to UNIFORM COST SEARCH except that this uses *g(n)* + *h(n)* instead of *g(n).*	✓ H. A* search	✓ H. A* search
___ is a simple recursive algorithm that attempts to mimic the operation of standard best-first search, but using only linear space.	✓ C. RBFS	✓ C. RBFS
____ expands nodes with minimal f(n).	✓ H. A* search	✓ H. A* search
____ is complete and optimal, provided that h(n) is admissible (for TREE-SEARCH) or consistent (for GRAPH-SEARCH).	✓ H. A* search	✓ H. A* search
____ expands nodes with minimal h(N). It is not optimal but is often efficient.	✓ D. Greedy best-first search	✓ D. Greedy best-first search
If h(n) is ___, then the values of f(n) along any path are nondecreasing.	✓ G. consistent	✓ G. consistent
__ is the simplest way to reduce memory requirements for A* is to adapt the idea of iterative deepening where the cutoff used is f(n) rather than the depth.	✓ A. IDA*	✓ A. IDA*
h(n) which is ___ is one that never overestimates the cost to reach the goal.	✓ I. admissible	✓ I. admissible
Straight-line distance (for example, in Romanian routing problem) is ___ because the shortest path between any two points is a straight line.	✓ I. admissible	✓ I. admissible
This ___ condition is also called sometimes monotonicity.	✓ G. consistent	✓ G. consistent
A heuristic h(n) is ___ if, for every node n and every successor n' of n generated by any action a, the estimated cost of reaching	✓ G. consistent	✓ G. consistent

the goal from n is no greater than the step cost of getting to n' plus the estimated cost of reaching the goal from n'.

All Answer Choices

A. IDA*

B. overestimating

C. RBFS

D. Greedy best-first search

E. linear

F. memory-efficient

G. consistent

H. A* search

I. admissible

Question 3

8 out of 10 points

Time and space complexity are measured in terms of
- *b:* maximum branching factor of the search tree
- *d:* depth of the least-cost solution
- *m* : maximum depth of the state space (may be infinite)
- l (letter L): is the depth limit.

Note: C* is the cost of the optimal solution path

Question	Correct Match	Selected Match
Time - Breadth-first search	✓ D. b^d (b to the power of d)	✓ D. b^d (b to the power of d)
Time - Uniform-Cost search	✓ E. b to the power of $(1+\lfloor C^*/\varepsilon \rfloor)$	✓ E. b to the power of $(1+\lfloor C^*/\varepsilon \rfloor)$
Time - Depth-First search	✓ A. b^m (b to the power of m)	✓ A. b^m (b to the power of m)
Time - Depth-Limited search	✓ B. b^l (b to the power of l)	✓ B. b^l (b to the power of l)
Time - Iterative-Deepening search	✓ D. b^d (b to the power of d)	✓ D. b^d (b to the power of d)
Space - breadth-first search	✓ D. b^d (b to the power of d)	✓ D. b^d (b to the power of d)
Space - uniform-cost search	✓ E. b to the power of $(1+\lfloor C^*/\varepsilon \rfloor)$	✓ E. b to the power of $(1+\lfloor C^*/\varepsilon \rfloor)$

Space - depth-first search	✓ D. b^d (b to the power of d)	✗ F. bm (b times m)
Space - depth-limited search	✓ C. bl (b times l)	✓ C. bl (b times l)
Space - Interative Deepening search	✓ D. b^d (b to the power of d)	✗ F. bm (b times m)

All Answer Choices

A. b^m (b to the power of m)

B. b^l (b to the power of l)

C. bl (b times l)

D. b^d (b to the power of d)

E. b to the power of (1+⌊C*/ε⌋)

F. bm (b times m)

Tuesday, October 6, 2015 4:52:00 PM CDT

← OK