LNMIIT/B.Tech./CSE/CORE/2018-19/ODD/AI/Q-I

LNMIIT
The LNM Institute of Information Technology

Artificial Intelligence (AI)

Quiz-I

Time: 30 minutes | Date: September 20, 2018 | Max. Marks: 40

Name: **Roll No.**

Read the following instructions carefully.

1. There are 8 questions printed on both sides. Make sure you have the complete quiz.

2. There can be ONE or MORE correct answers.

   - The **correct set** of answers awards you 5 marks
   - Leaving a question **un-attempted (i.e. empty set)** awards you 0.
   - Any **subset** of the correct set, whose cardinality is **one less than** the cardinality of correct set awards 2 marks.
   - Any set other than those mentioned above awards -1

   Example: If correct answers to a question are (a), (b) and (d), then encircling (a), (b), (d) awards you 5. Encircling (a)(b) or (a)(d) or (b)(d) awards you 2. Any other answer, including just (a), just (b) or just (d) or any other combination thereof awards you -1.

3. Please draw a circle around the answer of your choice using only a PEN not pencil.

4. Whenver in doubt, write your assumption and continue.

======xxxxxx======

1. In state space searching, we have defined two terms, *completeness* and *optimality*. We have been also considering the notions of time and space complexities. Choose the correct options:

   (a) BFS is optimal and complete in all the cases. It returns the shallowest solution, which is *always* the best solution. The time complexity of BFS is $\mathcal{O}(b^s)$, where s is the first level containing a goal state and b is the branching factor.

   (b) DFS is not optimal but it is *always* complete. The space complexity of DFS is $\mathcal{O}(bm)$, where b is the branching factor and m is the total number of levels.

   (c) UCS is both optimal and complete. However, it explores way too much than required and hence may end up exploring a lot of states, much more than required, to reach to the goal.

   (d) If a search algorithm is both optimal and complete, then the algorithm never returns any other sub-optimal goal before the best goal.

2. Consider the Romanian map problem that we had discussed in class. The start state is Arad and the goal state is Bucharest. Then which of the following are true:

   (a) There exists a carefully chosen heuristic h that makes A* same as UCS for the above problem.

   (b) h_1 is chosen as $\frac{\text{straight line distance}}{2}$ and h_2 is chosen as $\frac{\text{straight line distance}}{10}$. Another h_3 is defined as $\max\{h_1, h_2\}$. Then h_3 is also admissible.

   (c) Instead of using an admissible heuristic as we did in the class, we could have also used a consistent heuristic to solve the problem optimally.

   (d) There is no need to use admissible or consistent heuristic. We could solve the problem using Dijkstra's shortest path algorithm as well.

3. Optimality of A*

   (a) A* expands the nodes such that the f-value, given by, $f(n) = g(n) + h(n)$ increases. However, it need not maintain any fringe for being optimal.

   (b) A* expands the nodes such that the nodes with f-value, given by, $f(n) = g(n) * h(n)$ increases.

   (c) For nodes to be expanded in increasing $f(n)$ value, it is necessary that h should be admissible.

   (d) A* stops executing as soon as a goal state is *added* to the fringe.

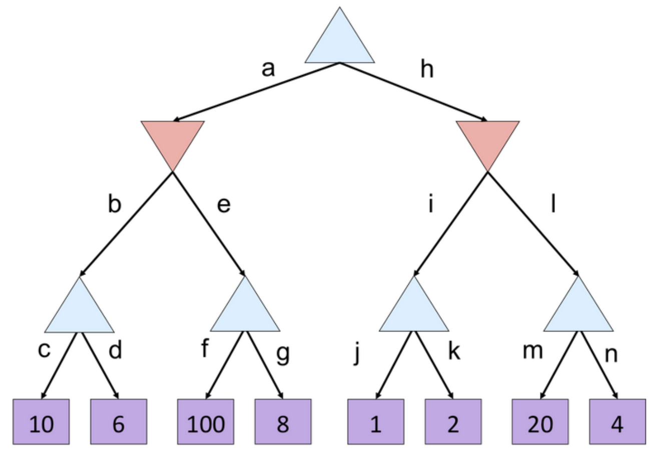
4. Graph vs Tree search
   (a) Graph search can be optimal and complete even if we do not maintain a fringe.
   (b) Admissibility and consistency are two measures to define the quality of a greedy heuristic h. If a heuristic is consistent, it is admissible too.
   (c) For graph search to be optimal, it is sufficient for heuristics to be admissible.
   (d) Graph search may not be complete because we are not traversing the whole graph.

5. Consider the min-conflict heuristic used for solving CSPs. We have seen that instead of starting from an empty assignment, min-conflict initializes the start state to a random state with complete assignment of variables.
   (a) Using this algorithm it is possible to solve n-queens very efficiently even for n=10 Million
   (b) The heuristic is to randomly select any conflicted variable and choose a value that violates the least constraints.
   (c) Min-conflict heuristic solves the CSPs efficiently for all the instances of the problem.
   (d) Critical ratio is that range of the ratio, for which nothing can be said about the status of the complexity of CSPs, i.e., there may not be any algorithm that solves the problem, efficiently or inefficiently.

6. Approaches to solve CSPs.
   (a) Since CSPs are a subset of the search problems and BFS is known to work well on search problems, it is a good idea to use BFS for CSPs as well.
   (b) *DFS with backtracking* was introduced because the worst case complexity of DFS alone is far worse than the worst case complexity of *DFS with backtracking*.
   (c) Minimum Remaining Values and Least Constraining Values are two heuristics to further improve upon *DFS with backtracking* and are known to work well in practice.
   (d) Arc consistency is a mechanism to find out inevitable failures early.

7. Consider adversarial games that are deterministic, two player and zero sum
   (a) The utility of the adversary is equal and opposite of the agent.
   (b) For games like Chess, it is easy to propagate the values of terminal utilities to the top. Hence, evaluation functions are not needed to approximate the values propagated by Minimax.
   (c) Zero sum property of the game ensures that the two players remain adversary throughout the game.
   (d) If the game is not zero sum, then there might be instances in the game when the players may cooperate.

8. Consider the above minimax tree. Assume that the adversary is optimal.



   (a) The best path in this given tree is $a \to b \to c$.
   (b) The terminal utility 100, even though the best sought after state for the player `max`, can never be achieved.
   (c) If the terminal utility 2 is replaced by 10 in the shown figure, then a lexicographic ordering may be sufficient to break the tie.
   (d) In the figure shown, replacing terminal utility 6 by 50 and replacing terminal utility 2 by 50 have the same impact on the propagation of values to higher levels.