Quiz-1, CSE 643: AI-SecB Full marks: 20 Duration: 40 mins 09-09-2025

Set-1

1. Choose True/False for the following statements. $1 \times 8 = 8$

   (a) BFS is a systematic and complete search algorithm. True

   (b) I want to prove that a path-finding problem is computationally intractable. For that, I can use polynomial-time reduction from a two-pair shortest-path problem. False

   (c) An approximation algorithm cannot find an optimal solution; it only finds a solution which is worse than the optimal solution by a factor of "approximation ratio". False

   (d) A* search is optimal for any heuristic function. False

   (e) Suppose the Depth-limited search algorithm needs to reach *one of the goals*. The algorithm increases depth-bound when $l < d$, where l is the initial depth-bound and d is the depth of the deepest goal. False

   (f) In Iterative Deepening A*, if a node is pruned it may not be pruned forever. True

   (g) A zero-sum game is a game where the total payoffs to all players is the same for every instance of the game. True

   (h) In a limited-horizon search with depth limit d, a surrogate goal node can be present at level $< d$. False

2. Answer the following:

   (a) Give a mathematical expression of approximation ratio of an approximation algorithm, with clear explanation of the terms. 1.5

   (b) Give an example of a graph for which Approx-Vertex-Cover algorithm, discussed in the class, always yields a suboptimal solution. Give reason. 1.5

   Solution:

   (a) An approximation algorithm for a problem has an approx ration $\rho(n)$ if:

   $$\max\left(\frac{C}{C^*}, \frac{C^*}{C}\right) \le \rho(n)$$

   , where C and $C*$ are the approximate solution and optimal solution respectively, and n is the input size of the problem.

   - If a student writes the above, give **full marks**.
   - If a student gives correct formulation for minimization and maximization problem individually, then given **full marks**.

- **Partial marks** for partially correct answer.

(b) For any complete graph with even number of vertices, the APPROX-VERTEX-COVER algorithm always yields a suboptimal solution.

- If a student writes the above, give **full marks**.
- If a student gives an example of a complete graph with even number of vertices, then also give **full marks**.
- If a student gives a different example which is verified to be correct, give **full marks**.
- **No marks** for wrong answer.

3. A directed graph $G = (V, E)$ is *singly connected* if $u \rightsquigarrow v$ implies that G contains at most one simple path from u to v for all vertices $u, v \in V$. Give an efficient algorithm to determine whether or not a directed graph is singly connected. 3

Solution: See Algorithm 0.1.

Algorithm 0.1: Checks if a graph is singly connected or not.

Input: Graph $G = (V, E)$
Output: TRUE or FAILURE

function Is-Singly-Connected(G)
begin
    for $i \leftarrow 1$ **to** $|V|$ **do**
        Initialize a set $S_{v_i} \leftarrow \emptyset$ for vertex v_i
        for *each vertex* v_i **do**
            Start BFS or DFS from v_i
            for *each visited node* v_j **do**
                if $v_j \in S_{v_i}$ **then**
                    return FALSE; G is not singly connected.
                end
                $S_{v_i} \leftarrow S_{v_i} \cup \{v_J\}$
            end
        end
    end
    return TRUE; G is singly connected.
end

- If a student writes something similar to the following or any other algorithm which is correct give **full marks**.
- If the answer is written in algorithmic structure or explained in words that ensures correctness, give **full marks**.

- Additionally, if a student proposes a method that reuses previously computed sets (or lists), give a bonus mark of 1 on top of the full marks.
- **Partial marks** for partial correctness.

4. In A* algorithm, consider a heuristic function h that obeys the *consistency* condition. Consider a pair of nodes n_i and n_j in a graph such that n_j is a successor of n_i. Here, the terms f, g, h have usual meaning. Answer the following:

   (a) Write a mathematical expression between $h(n_i)$ and $h(n_j)$. 1

   (b) Prove that the f-values of the nodes in the search tree are *monotonically non-decreasing* as we move away from the start node of the graph. 2

   Solution:

   (a) As per the *consistency* condition of h, for two nodes n_i and n_j:

   $$h(n_i) - h(n_j) \leq \texttt{dist}(n_i, n_j) \tag{1}$$

   where $\texttt{dist}(n_i, n_j)$ is the actual (shortest) distance between n_i and n_j.

   (b) As n_j is a successor of n_i, we have:

   $$\begin{aligned} h(n_j) &\geq h(n_i) - \texttt{dist}(n_i, n_j) && \text{[from (1)]} \\ h(n_j) + g(n_j) &\geq h(n_i) - \texttt{dist}(n_i, n_j) + g(n_j) && \text{[adding } g(n_j)\text{[]]} \\ f(n_j) &\geq h(n_i) - \texttt{dist}(n_i, n_j) + g(n_i) + \texttt{dist}(n_i, n_j) && \text{[as } g(n_j) = g(n_i) + \texttt{dist}(n_i, n_j)\text{]} \\ f(n_j) &\geq f(n_i) && \text{[as } f(n_i) = g(n_i) + h(n_i)\text{]} \end{aligned} \tag{2}$$

   - If a student uses Eq. 1 give him/her 1 .
   - Give marks according to correctness of the rest of the steps.
   - If Eq. 1 or similar expression not used, give 0.

5. Answer the following:

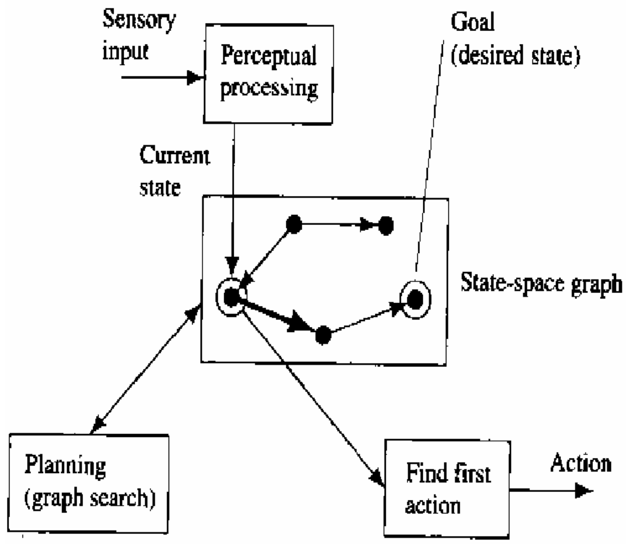
   (a) Draw a schematic diagram for the Sense-Plan-Act architecture. 1.5

   (b) Consider a two player game where one is maximizing and other is minimizing player. Given a state and a player, write a function to calculate *minimax value* for each player. 1.5

   Solution:

   (a) The diagram should like the following:

   - Please refer to **Section 10.1 of Book–Nilsson** before you start evaluation.



- Give full marks if the connection between perception (sensing), planning and action is shown via state-space graph.
- Deduct marks if any of the four components is missing or the connection is not proper.

(b) The function is as follows:

$$\textsc{Minimax}(s) = \begin{cases} \textsc{Utility}(s) & \text{if } \textsc{Terminal-Test}(s) \\ \max_{a \in Actions(s)} \textsc{Minimax}(\textsc{Result}(s,a)) & \text{if } \textsc{Player}(s) = \textsc{max} \\ \min_{a \in Actions(s)} \textsc{Minimax}(\textsc{Result}(s,a)) & \text{if } \textsc{Player}(s) = \textsc{min} \end{cases}$$

- Please refer to the introduction of **Section 5.2 of Book–Russell-and-Norvig** before you start evaluation.
- Give full marks only if they mention what the other functions (such as `Result` and `Terminal`*do*.).
- In case they properly convey the concept, give full marks.
- Deduct marks if any of the three conditions are not mentioned.
- No marks if they do not define any function.

*** Thank you ***