### Class 21BIT – Term II/2023-2024

### Course: CS300 – Artificial Intelligence

Homework 02

***Submission Notices:***

* *Conduct your homework by filling answers into the placeholders given in this file (in Microsoft Word format). Questions are shown in black color, instructions/hints are shown in italic and blue color, and your content should use any color that is different from those.*
* *After completing your homework, prepare the file for submission by exporting the Word file (filled with answers) to a PDF file, whose filename follows the following format,*

*<StudentID-1>\_<StudentID-2>\_HW02.pdf (Student IDs are sorted in ascending order)*

*E.g.,* ***2152001\_2152002\_HW02.pdf***

*and then submit the file to Moodle directly WITHOUT any kinds of compression (.zip, .rar, .tar, etc.).*

* *Note that you will get zero credit for any careless mistake, including, but not limited to, the following things.*
  1. *Wrong file/filename format, e.g., not a pdf file, use “-” instead of “\_” for separators, etc.*
  2. *Disorder format of problems and answers*
  3. *Conducted not in English.*
  4. *Cheating, i.e., copying other students’ works or let the other student(s) copy your work.*

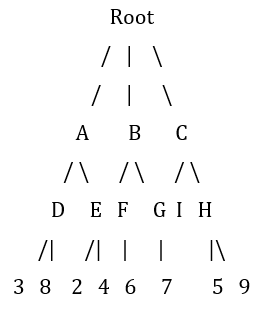
**Problem 1. (2pts)** Please answer the following questions by filling in the blanks on the right side of the below table.

|  |  |
| --- | --- |
| **Questions** (0.5pt each) | ***Filling in the blanks*** |
| What is a “constraint” in a Constraint Satisfaction Problem (CSP)? | *Answer: A solution in a Constraint Satisfaction Problem (CSP) involves a “constraint” that limits the available values or configurations of a variable to solve the issue. These constraints specify the connections between variables and must be followed for a legitimate solution, narrowing down possible possibilities.* |
| In the context of game-playing, how does the Minimax algorithm utilize adversarial search, and what role does the Alpha-Beta pruning technique play in optimizing this search process? | *Answer: The Minimax algorithm is a decision-making algorithm used in two-player games like chess or tic-tac-toe, where one player maximizes their score and the other minimizes it. It uses adversarial search, awarding points to potential moves and choosing the highest score for the maximizing player.*  *Alpha-Beta pruning is a strategy used to improve the Minimax algorithm by reducing the number of nodes assessed in the game tree. It involves keeping two numbers, alpha and beta, indicating the maximum and minimum player scores. If a branch is discarded, the algorithm prunes it to save computing time, making the Minimax method more effective for complex game trees.* |
| What are the main limitations of the hill-climbing technique, and how might they be addressed or mitigated? (describe at least two) | *Answer:*  *Local Optima: The algorithm, Hill-climbing, often finds itself in Local Optima, failing to locate the global optimum. To overcome this, random restarts and simulated annealing are employed. Random restarts involve repeatedly retrying the algorithm with different solutions, increasing the chances of escaping Local Optima. Simulated annealing reduces premature convergence to Local Optima. Plateau Problem: The algorithm in Hill-climbing faces challenges in selecting the best solution when encountering plateaus with multiple equally good solutions. To mitigate this, randomization can be introduced. By randomly choosing among neighbors that are better or equal to the current solution, the algorithm can explore different directions, potentially leading to a better solution.* |
| How does Expectimax differ from Minimax? | *Answer:*  *Minimax: Minimax is a deterministic decision-making method that assesses the tree of possible movements by weighing the best and worst outcomes for each player, switching between maximizing, and minimizing players, suitable for games requiring comprehensive information.*  *Expectimax: Expectimax is a method that considers the probabilityistic nature of a game, evaluating the average expected utility by considering not only optimal moves but also the probabilistic outcomes. It is commonly used in scenarios where elements are random or against non-optimal opponents.* |

**Problem 2. (1pt)** In the plot, if we used hill climbing, which starting point would lead to a suboptimal solution? Draw a vertical line from your point to the x-axis and answer *‘possible’* in the bottom cell. If you cannot identify such a point, write *‘not possible’*.

|  |  |
| --- | --- |
|  |  |
| Answer: Any point that would not lead to the highest peak would suffice. | Answer: Every starting point would lead to an optimal solution, because this is a convex function. |

**Problem 3. (2pts)** Let's consider a simplified game scenario with a game tree for you to evaluate using the Minimax algorithm. This will be a hypothetical two-player game, where Player X is the maximizing player, and Player Y is the minimizing player. Play X plays first at the root.



Starting from the leaves:

* Node D has values [3, 8]
* Node E has values [2, 4]
* Node F has values [6]
* Node G has values [7]
* Node H has values [5, 9]
* Node I has NO any children

1. **(1pt)** Determine the values for nodes A, B, C, and the Root.

Answer: A=4, B=6, C=9, Root=9

1. **(1pt)** Assuming an alpha value of 4 and a beta value of 6 at the root when starting the exploration. At which node (if any) can you apply Alpha-Beta pruning, and why?

Explanation:

G node ( because the alpha value and beta value at node B is 6=6, which made the G node being cut off) and C, I and H node cause by the value of alpha and beta in Root node(6=6)

**Problem 4. (2pts)** An, Binh, Cuong, and Dang are choosing their warriors for the Arena Of Valor game. The possibilities are Tulen, Zuka, Natalya, and Violet. Each of them have the following preferences:

1. Cuong will not pick Violet.
2. An and Binh want to steal each other’s warrior, and thus it is pointless if they both choosing the same warrior. So, An and Binh will pick different warriors.
3. Binh likes ability power warriors, so he will only choose Tulen or Natalya.
4. Cuong likes to be unique in his preferences and will not choose the same warrior as anybody else except An. Since An and Cuong are twins, they always choose the same warrior.
5. Dang really dislikes Zuka, and hence he will not play that warrior.

*Please write your answer in the table*

|  |  |
| --- | --- |
| Draw the corresponding constraint graph for this CSP. (0.5pt) | Edges: A-B, C-A, C-B, C-D |
| We will run basic backtracking search to solve this CSP and make sure that every person (variable) is matched with their desired warrior (value). We will select unassigned variables in alphabetical order, and we will also iterate over values in alphabetical order. What assignment will backtracking search return? (0.5pt) | An: Natalya  Binh: Tulen  Cuong: Natalya  Dang: Tulen |
| We will run one iteration of forward checking. Assume that no values have been assigned to the variables. Write down the value(s) that will be eliminated if we assign “Tulen” to An. Write down “None” if no values will be eliminated. (1pt) | Values that will be eliminated for Binh: Tulen  Values that will be eliminated for Cuong: Zuka,  Natalya, Violet  Values that will be eliminated for Dang: None |

|  |  |
| --- | --- |
| **Problem 3. (2pts)** Consider the following heuristics for the 8-puzzle problem, in which the goal state is fixed as shown aside.   * hA(n) = Number of tiles out of row + Number of tiles out of column * hB(n) = P(n) + 3\*S(n) |  |

* + P(n): Sum of Manhattan distances of each tile from its proper position
  + S(n): A sequence score obtained by checking around the non-central squares in turn, allotting 2 for every tile not followed by its proper successor and 0 for every other tile, except that a piece in the center scores 1.

For example,

|  |  |  |
| --- | --- | --- |
|  | hA(n) = 1 + 1 = 2  hB(n) = 2 + 3\*5  = 17 | * Tiles out of row: 1; Tiles out of column: 1 * P = 2 * S = 1 (tile 2) + 2 (tile 1) + 0 (tile 3) + 0 (tile 4) + 0 (tile 5) + 0 (tile 6) + 0 (tile 7) + 2 (tile 8) = 7 (clockwise) |

Prove the admissibility of the two given heuristics.

|  |  |  |
| --- | --- | --- |
| **Heuristics** (1pt each) | **Admissible?** | **Explanations** |
| hA(n) | This heuristic is admissible | Every tile that is out of column or out of row must be moved at least once, and every tile that is both out of column and out of row must be moved at least twice. Therefore, the total number of moves will always be equal to or greater than hA(n), making it a lower bound on the cost and thus admissible. |
| hB(n) | This heuristic is not admissible | A heuristic is **non-admissible** if it can overestimate the cost to reach the goal from a given state. This is contrary to an admissible heuristic, which never overestimates the cost. |