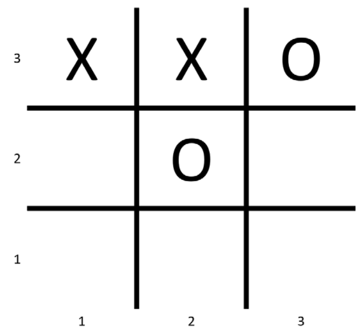
CPSC 4420/6420 Artificial Intelligence
Assignment 3.

Due on Tuesday, 11/07 at 11:59pm

Problem 1: Game Tree Visualization and Evaluation: (15 points)

Part 1) Consider the following Tik-Tak-Toe game.



a) It is O’s turn. Sketch the game tree from this position [state], showing all possible moves for O, followed by X's responses. No need to determine the best action.

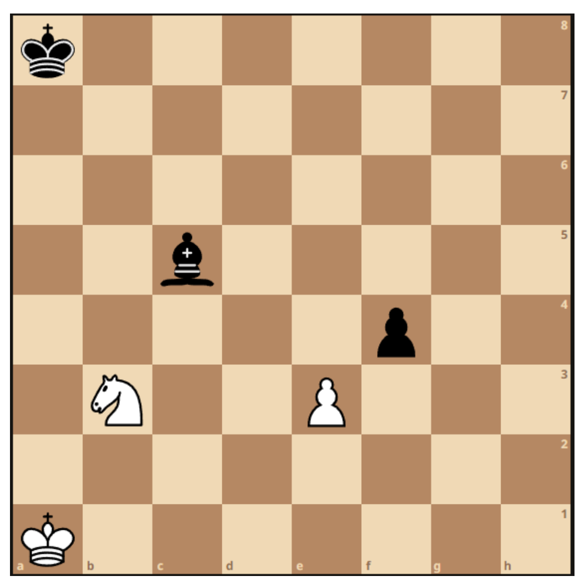
Part 2) Consider the following scenario: Imagine you are in the middle of a chess game. The board only contains the following pieces:

White: One Knight at B3, one Pawn at E3, one King at A1

Black: One Bishop at C5, one Pawn at F4, one King at A8

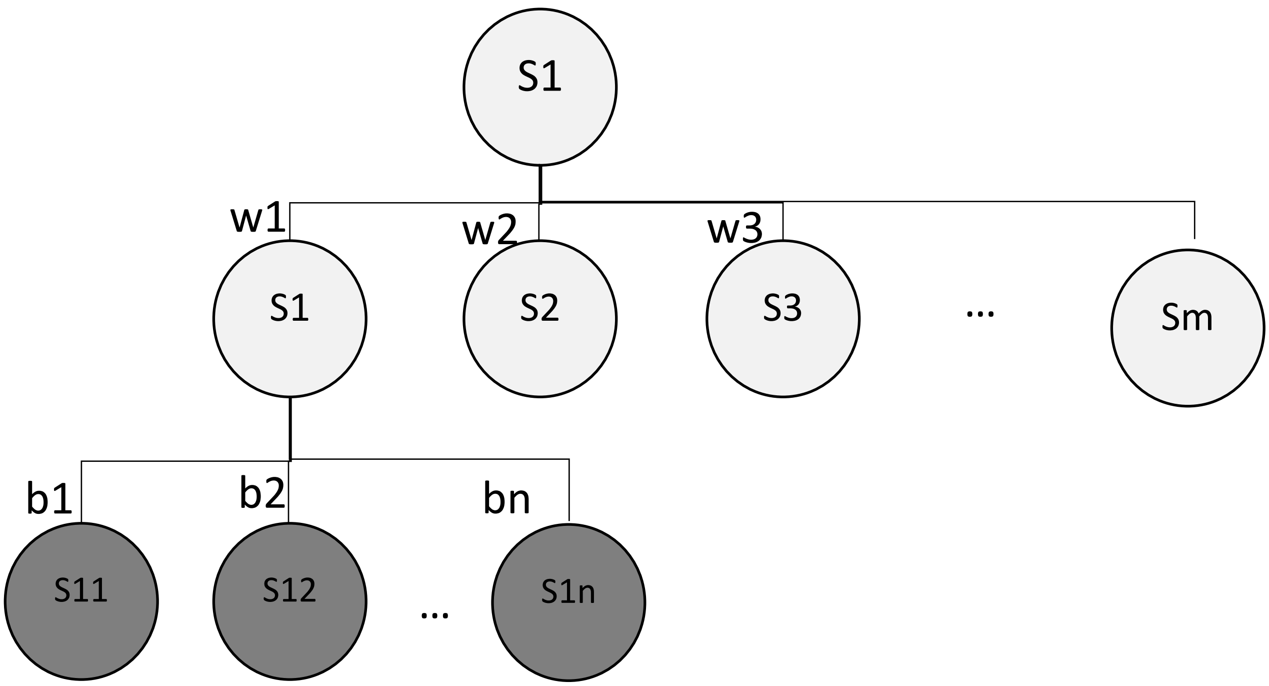
Let’s show this initial state as

S0= (WK:A1, WP: E3, WKn: B3, BK:A8, BP: F4, BB: C5)



It's White's turn.

b) Draw the game tree that shows all possible moves for White from this state, and the select one of the states by your choice and present all possible responses from black. You can represent states as follows and you don't need to draw the chess board.



Hint:

Example:

w1: WK:A1->A2   S1: (WK:A2, WP: E3, WKn: B3, BK:A8, BP: F4, BB: C5)

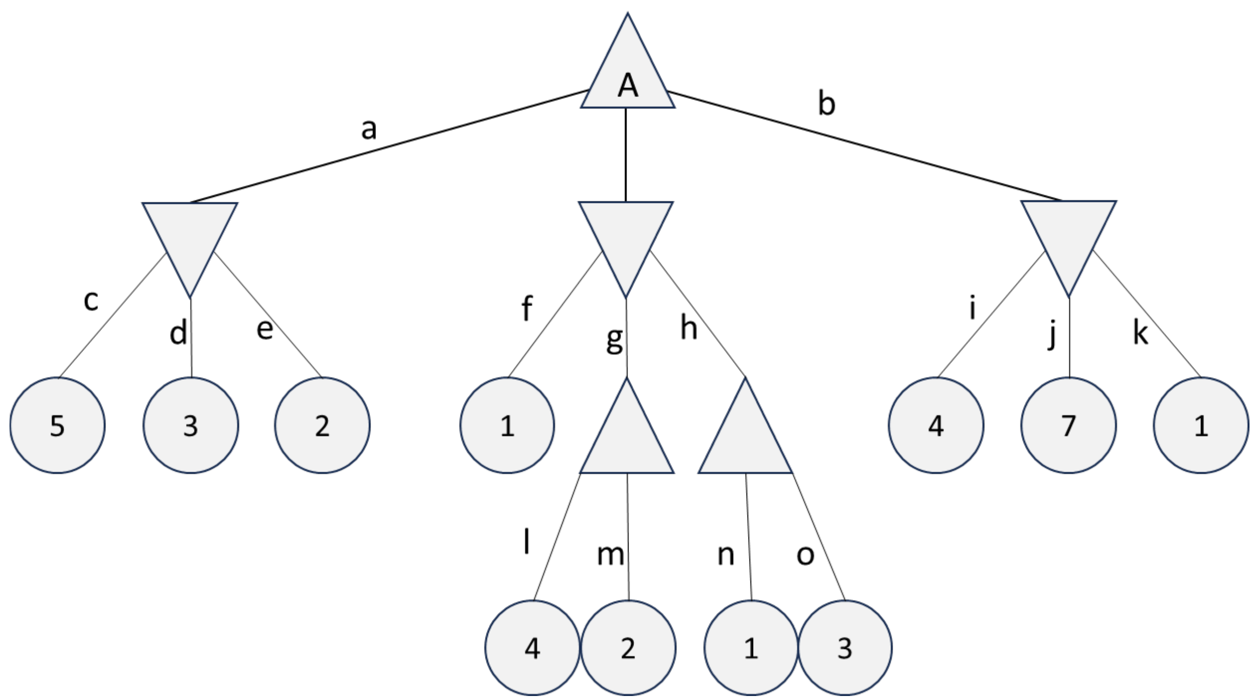
w2: WK:A1->B2   S1: (WK:B2, WP: E3, WKn: B3, BK:A8, BP: F4, BB: C5)

…

c) Using the assigned point values: Pawn (1), Knight (3), Bishop (3), Rook (5), Queen (9), and King, assess the worth of each resulting game state following White's move (S1, S2, ... Sm as mentioned in part 2b) by calculating the difference between the total value of White's pieces and the total value of Black's pieces. Identify which state holds a material advantage. Does this assessment align with your intuition?

Problem 2: Min-Max and Alpha-Beta Pruning: (10 points)

Consider the following tree:



a) Using the Min-Max algorithm, determine the best move for A and what is the resulting reward value. Write all the steps, like the way you did in Quiz.

b) Repeat the problem using Alpha-Beta pruning. Indicate which branches are pruned. Write all the steps, like the way you did in Quiz.

c) Change min nodes with EXP nodes and repeat part a. (execute Expectimax algorithms).

Question 3: Expectimax Algorithm: (20 points)

Game: Two-Player Dice Showdown

Rules:

1. Players A and B take turns to roll a die.
2. Each player, on their turn, has two choices:
    - Roll a 6-sided die
    - Roll a 4-sided die and add 1 to their total score.
3. Each die roll adds the resulting number to the player's total score for the game.
4. The game ends when a player reaches or exceeds a score of 10. The first player to do so wins.
5. If Player B reaches a score of 10 on their turn, they automatically win, even if Player A has not had an equal number of turns.

Question:

Player A has a total score of 8, and it's their turn. Player B has a total score of 7.

a) Sketch or describe the decision tree for Player A's turn, including expectation nodes.

b) Using the Expectimax algorithm, determine which action (rolling a 6-sided die or rolling a 4-sided die and adding 1) maximizes Player A's chance of winning the game on this turn. Show your calculations.

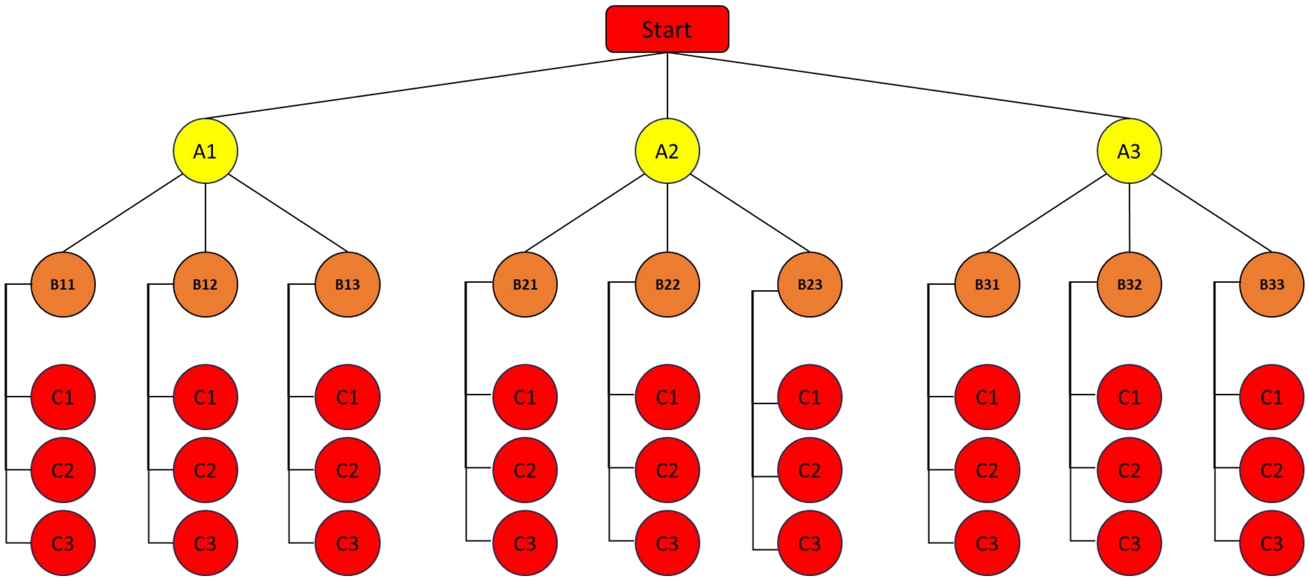
c) If Player A does not win on this turn, sketch or describe the decision tree for Player B's subsequent turn, including max, min, and expectation nodes.

d) Using the Expectimax algorithm, determine Player B's optimal strategy and Player A's expected utility given Player B's optimal strategy.

Problem 4: Multi-player adversarial search, the triad tournament (25 points)

Scenario: Imagine a triangular board game called Triad. In Triad, three players – A, B, and C – compete in turns, each aiming to maximize their individual scores. The game progresses sequentially: Player A moves first, followed by Player B, and then Player C. After Player C, it circles back to Player A.

Given a partial game tree, the gameplay unfolds as below:



- The leaf nodes, representing Player C's moves, have terminal states with values denoting points for players A, B, and C. For instance, a leaf node with the value (5,3,7) means Player A scores 5 points, Player B scores 3 points, and Player C scores 7 points for that particular sequence of moves.

- Terminal state values for C's moves are as follows:

  A1 -> B11: [(5,3,7), (4,4,5), (3,6,3)]

  A1 -> B12: [(2,5,6), (6,2,5), (4,5,4)]

  A1 -> B13: [(3,4,8), (5,3,6), (2,6,7)]

  A2 -> B21: [(4,5,6), (5,4,6), (3,5,7)]

  A2 -> B22: [(3,6,6), (4,3,8), (5,5,5)]

  A2 -> B23: [(5,2,8), (6,3,4), (4,4,7)]

  A3 -> B31: [(6,3,6), (5,4,6), (4,5,6)]

  A3 -> B32: [(3,5,7), (4,6,5), (5,5,5)]

  A3 -> B33: [(5,3,7), (6,2,7), (4,5,6)]

a) Using the terminal state values provided, work backwards:

   1) Update the values at Player B's decision nodes, determining the best possible choices for Player B (for all plays that A may have done).

   2) Once Player B's nodes have values, update the values at Player A's decision nodes, determining the optimal move for Player A considering Player B's optimal moves.

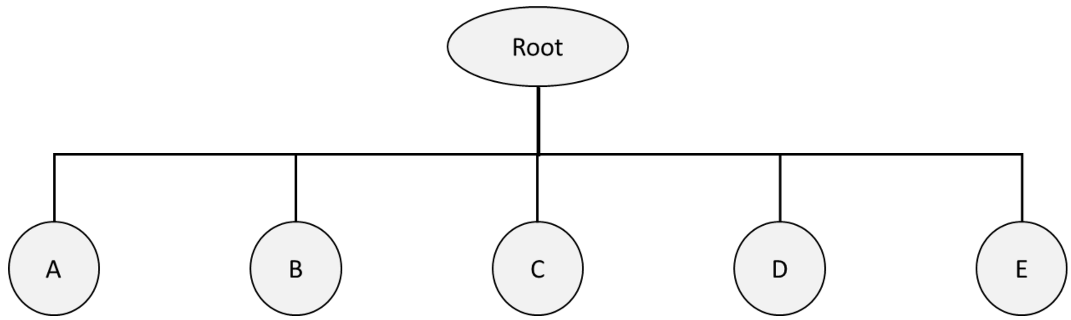
b) Document the best starting move for Player A based on the updated values.

c) For Player A's optimal move, determine the best countermove by Player B.

d) Considering the chosen moves of Players A and B, pinpoint the best moves for Player C for each scenario.

Problem 5: Mote Carlo Tree Search. (30 point)

You're building a board game AI using MCTS. The current game state can branch out into five possible moves, represented by nodes A through E. Below are the current statistics of each node:

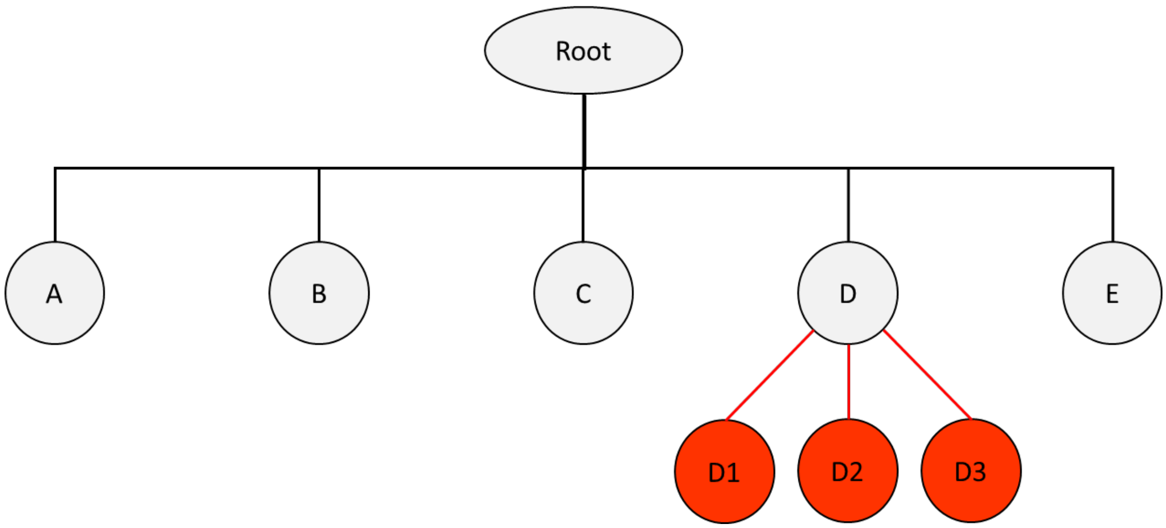


Given:

- **Visits to the Root (N)**: 100
- **Node A (n_A)**: Visits = 21, Value = 10 (value is the number of wins over plays v(A)=10/21)
- **Node B (n_B)**: Visits = 25, Value = 13 (x/y: 13/25)
- **Node C (n_C)**: Visits = 17, Value = 9 (9/17)
- **Node D (n_D)**: Visits = 27, Value = 18 (18/27)
- **Node E (n_E)**: Visits = 10, Value = 6 (6/10)

a) Calculate the UCB value for each node (A through E) for C= 1 and specify which node is selected.
UCB (state) = value + C * sqrt [log_e (sum of plays of the parent) / # of play)]
b) Repeat part a for C = 10, and specify which node is selected this time.
c) Describe how the choice of the exploration constant C impacts the trade-off between exploitation and exploration. What might happen if C is set very high? What about very low?

- Continuing from the previous game state, suppose Node D is selected for expansion. It has three child nodes: D1, D2, and D3. We ran 9 simulations from these nodes that yielded the following results:



D1: Two wins (+1 each) and one loss (-1).

D2: One win (+1) and two losses (-1 each).

D3: Three losses (-1 each).

d) Detail the backpropagation process to update the values and visits for Node D, its child nodes (D1, D2, D3), and the Root after these simulations. You can notation x/y where x is the number of wins and y is the number of plays.

e) Repeat the selection process for the new tree (nodes A,…E, and D1,D2,D3) using C= 0.25. Explain how the results of the simulations affect the future selection of child nodes under Node D during the next MCTS iterations.