Artificial Intelligence

CT-21003

Theory Assignment – Adversarial Game

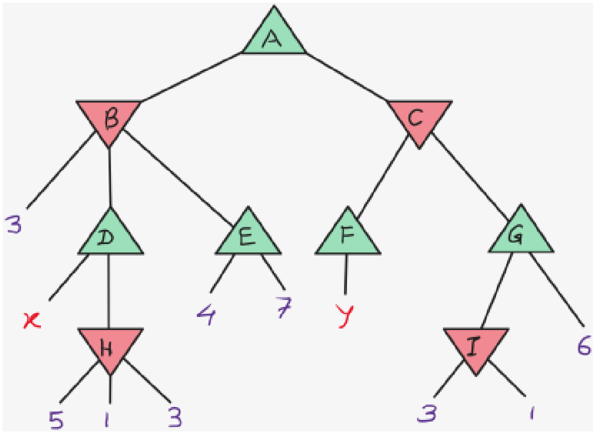
Q1. Consider the following Game:

A			B
1	2	3	4

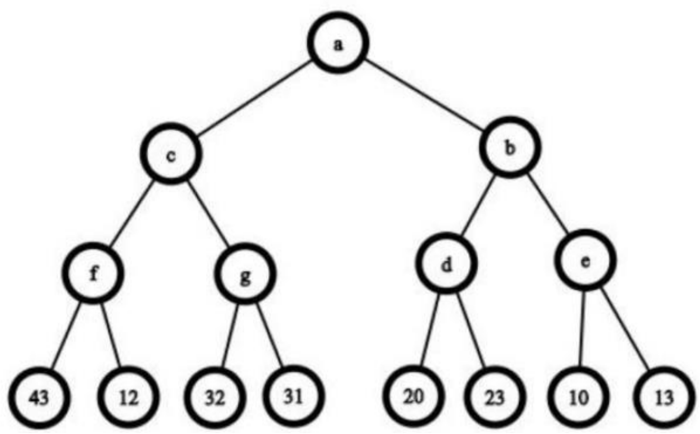
The starting position of a simple game as given above. Player A moves first. The two players take turns moving, and each player must move his token to an open adjacent space in either direction. If the opponent occupies an adjacent space, then a player may jump over the opponent to the next open space if any. (For example, if *A* is on 3 and *B* is on 2, then *A* may move back to 1. The game ends when one player reaches the opposite end of the board. If player *A* reaches space 4 first, then the value of the game to A is +1; if player B reaches space 1 first, then the value of the game to A is −1.

1. Draw the complete game tree, using the following conventions:
   - Write each state as (s_A, s_B), where s_A and s_B denote the token locations.
   - Put each terminal state in a square box and write its game value in a circle.
   - Put loop states (states that already appear on the path to the root) in double square boxes. Since their value is unclear, annotate each with a "?" in a circle.
2. Now mark each node with its backed-up minimax value (also in a circle). Explain how you handled the "?" values and why.
3. Explain why the standard minimax algorithm would fail on this game tree and briefly sketch how you might fix it, drawing on your answer to (b). Does your modified algorithm give optimal decisions for all games with loops?
4. This 4-square game can be generalized to n squares for any $n > 2$. Prove that *A* wins if n is even and loses if n is odd

Q.2 Consider the following tree, where max and min agent are shown as green and red color triangles. Two leaves x and y are not specified. What is the lower bound (at least) and upper bound (at most) value of x that makes the minimax value of node A to be x by considering y = 0?



Q3. Trace the execution of alpha-beta algorithm for the following given game tree. Show the min-max values for each of the non-terminal nodes assuming children are visited in left-to-right order.



Q4. Trace the execution of alpha-beta algorithm for the game tree given in question 3. How many nodes will **not be visited** by alpha-beta pruning, assuming children are visited in left-to-right order.