Name - Vasu Kalariya
Roll - PE29
Sub - AI

Lab Assignment -2

Title: Implementation of MinMax algorithm for Tic-Toc-Toe game.

Aim: Solve Tic-Toc-Toe using MinMax algorithm

Objective: To study & implement minmax algorithm for Tic Toc Toe

Theory

→ Adversarial search:
Adversarial search is a search when there is an enemy or opponent changing the state of the problem every step in a direction you do not want.
Eg: Chess, business, trading, war.
You change state, but then you don't control next state opponent will change next state in a way.
a) Unpredictable
b) hostile to you
You can get to change every alternate state

→ Tic-Toc-Toe solving steps.
Consider two opponents, 1st represent by X & the other by 'O' where we aim on maximizing the chance of 'X' winning. Rules are as follow.
a) If X wins, take it.
b) If opponent wins, block it
c) If possible create a fork (2 wining ways)
d) Do not let opponent block 'X' winning move
e) If neither 'X' or 'O' wins call it a tie.

→ Data Structure & other details about MinMax algo.
MinMax is a backtracking algo. that is used in decision making & game theory to find optimal more for a player. A Binary tree is used for this algo. It has 2 players ~~maxmin~~ maximizer who tries to get highest ~~so~~ score possible & minimizer who tries to get highest score possible. It is widly used in 2 player turn-based games such as tic-toc-toe Backgmmon, mancala, Chess, etc. Performs depth-first search algorithm

Input : Initial State

Output: Solution / goal state with optimal path

Algorithm: Min Max

FAQ

1 Compare Informed search & adversarial search.

Informed search	Adversarial search
→ Uses knowledge for search process	→ Doesn't use knowledge for searching process.
→ Finds solution quickly	→ Finds solution slowly
→ Cost is low	→ Cost is height
→ It takes less time	→ It takes moderate time
→ less lengthly while implementation	→ more lengthy while implementation.

2 Explain minimax algorithm with example.

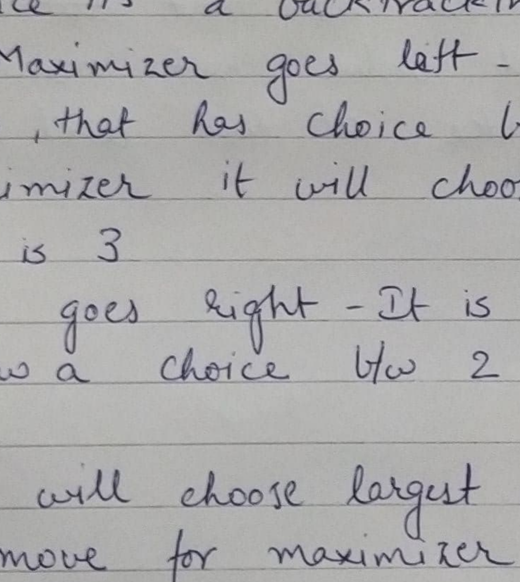
Every board state has value associated with it. In a given state if maximizer has upper hand then ~~so~~ score of board will tend to be some positive value if minimizer has upper hand in that board state then it will tend to be some ~~neg~~ negative value. Values of board are calculated by some heuristics which are unique for every type of game.

Eg: Consider game with 4 final states & maximicing player starting first. The game tries all possible moves since its a backtracking algo.

~~Maximum~~ Maximizer goes left - it is minimizer turn now, that has choice betwⁿ 3 & 5
Being minimizer it will choose ~~last~~ least among both that is 3

Maximizer goes right - It is minimizer turn. It has now a choice b/w 2 & 9 He will choose 2

Maximizer will choose largest value = 3. Hence optimal move for maximizer is to go left



3 Explain Alpha beta pruning.

It is a optimization technique for minimax algorithm. It reduces computation time & allows us to search much faster & even go into deeper levels in game tree. It cuts off branches in game ~~tree~~ tree which need not be searched because there already exists a better move available. It passes 2 extra parameters in minimax function

Alpha - Best value that maximizer currently can guarantee at that ~~to~~ level or above
Beta - Best value that minimizer currently can guarantee at that level or above.