Name : Alok Bhawarikar
Roll : PA06
Panel : 1

Lab Assignment 2 — AI

Title : Implantation of minimax algorithm for Tic-Tac-Toe game.

Aim : Solve Tic-Tac-Toe using minimax Algorithm

Objective : To study and implement minimax algorithm for Tic-Tac-Toe

Theory :

Adversarial Search :

- examine the problem contain we try to plan ahead of world and other agents plans against me.
- Searches in which 2 or more players with conflicting goals are trying to explore same space for solution.
- "opponent" changing state of problem every step in a direction we do not want.
- eg. chess, Tic-Tac-Toe.
- includes - minimax algorithm
  - alpha beta Pruning

Tic - Tac - Toe

- 2 Players x & o
- turn-by-turn play in 3x3 and
- 3 in a row, column or diagonal first wins.
- Considering minimax algorithm

- 2 players max and min.
- players have an alternate turn & shirt with max.

MAX → maximum result of game here
Min → minimise result

- Utility function f(p) 'p' be position / state in game

F(p) = { largest positive number, if p is win for computer.
Smallest negative number, if p is win for opponent RCDC − RCDO.

RCDC = no. of rows, columns, diagonals, in which computer would still win.

RCDO = no. of rows, columns, diagonals in which opponent could still win.

O	O	X
X	X	O
X	O	X

__Minimum Algorithm__

- recursive / backtracking algorithm used in decision making & game theory.
- provides optimal move for player assums opponent playing optimally.
- 2 players play to get maximum benifit, max & min

- Max will select maximum value, min selects minimized.
- performs 'DFS' for exploration of complete game tree.
- It proceeds always down to terminal node of tree, then back track the tree as recursion.
- Minimax stategy:
  - MIN value taken every other level of tree which represent opponent's choice of move.
  - computer assumes that human will choose move, of least value to computer

Properties:
- Complete, optimal.
- T.C & S.C is O (b^m) m is max depth of tree

Limitations:
- gets slow for complex games.

Input : Initial state.

Output : Solution / goal state with optimal path.

Algorithm : Minimax

Platform : Linux

FAQs

1) Compared infotmed search and adversanal search.

A) Informed search.

- Problem - specific knowledge that helps agents to esxplore less to search space and find more efficiently goal node.
- Uses idea of heuristics to identify most promisin search path.
- includes BFS & A* algorithm.

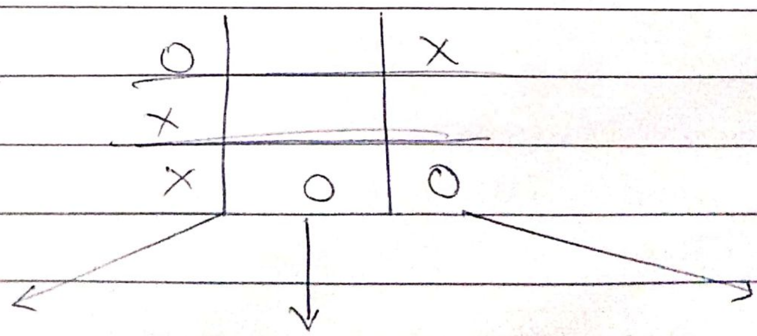
Adversanal Search.

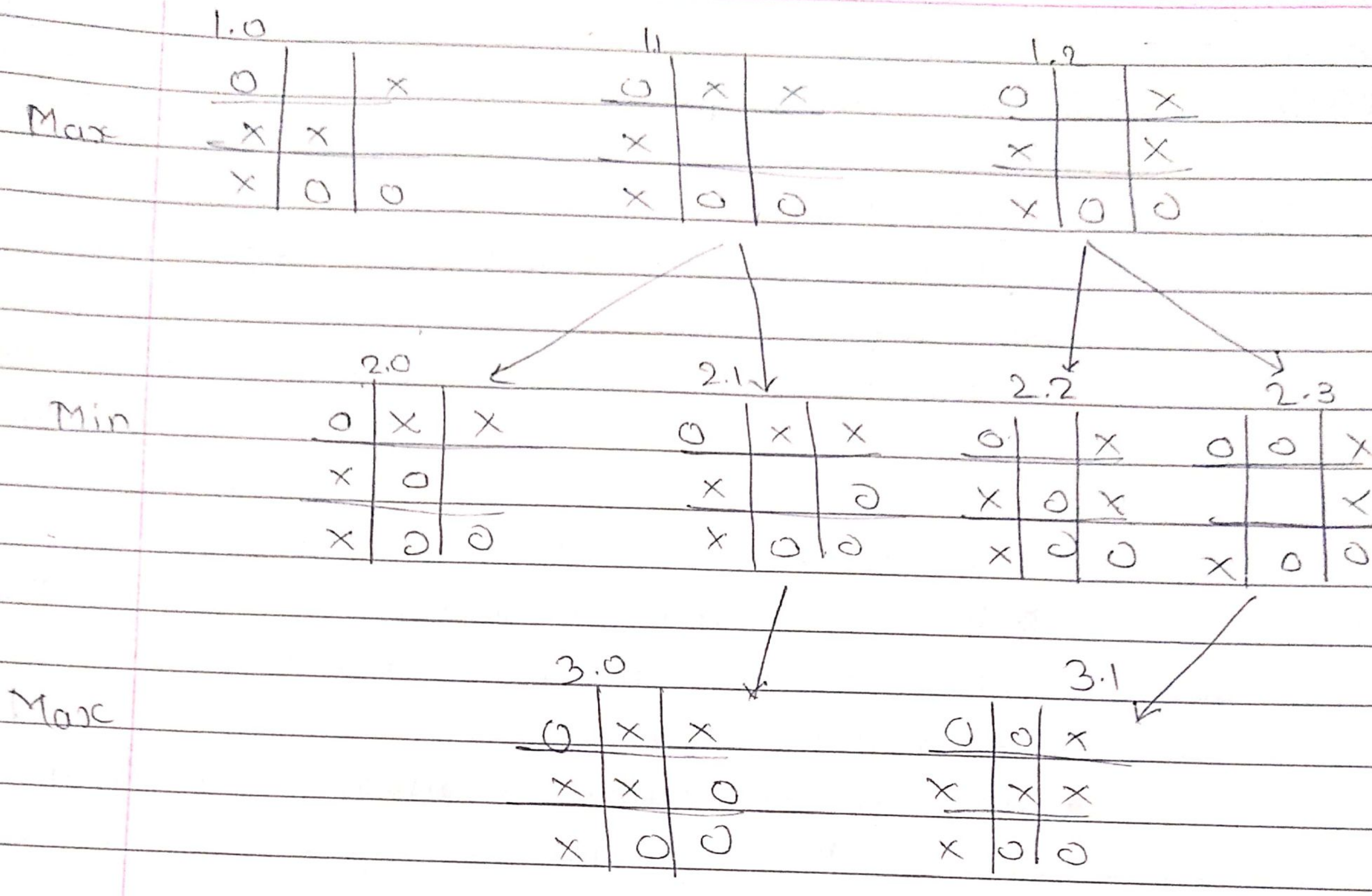
- escamines problems which are when we try to plan ahead of world & others agents planning against us.
- include multi-agent environment.
- includes games like chess.

Q2] Explain minimax algorithm with escample.

A) - recursive / back tracking algorithm used in session making & game theory.

eg :





- Initial state 0.0 gives 3 possibilities (1.0, 1.1, 1.2)
- 1.0 gives computer win (+1)
- 1.1 gives opponent 2 possibilities (2.0, 2.1)
  - 2.0 is winning state for our opponent
  - 2.1 gives possibilities, 3.0 in which computer is wing (+1)
- 1.2 gives 2 possibilities (2.2, 2.3)
  - 2.2 is winning state for opponents. so losing for computer is wing (+1)

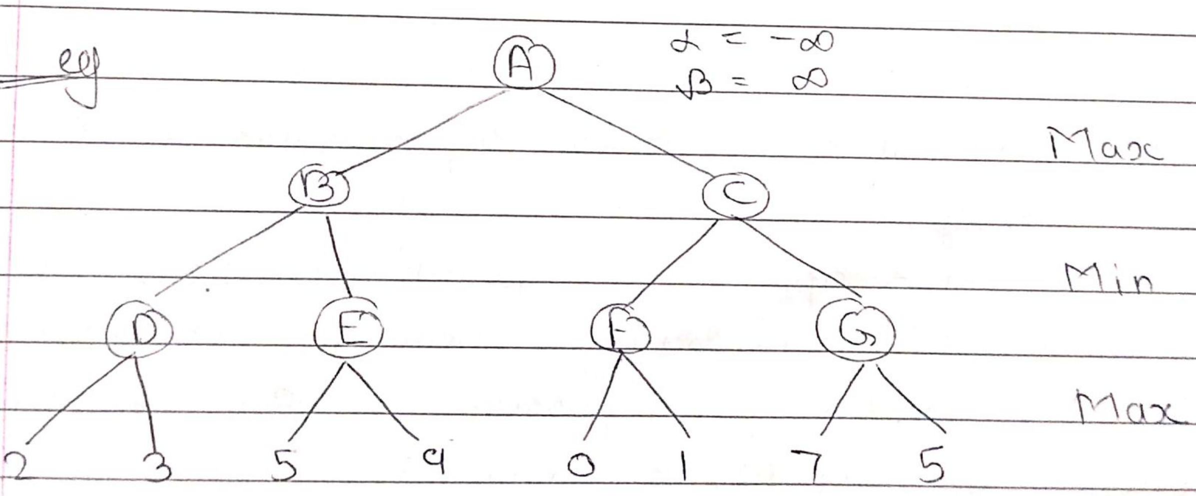
Q3] Explain Alpha Beta penity

A) Alpha Beta Pruning

- modified version of minimum & optimization technique
- correct minimax decision without checking each node of game tree has 2 threshold parameter alpha & beta.
- main condition for prenity is $\alpha \geq \beta$.
- return same move at regular, removes all nodes which are not really affecting final decision but many algo slow

* max player update & alpha values & min player beta values.

eg



1] Max player will start for Root A where $\alpha = -\infty$ & $\beta = \infty$ and same value passed to to its child and subchild.

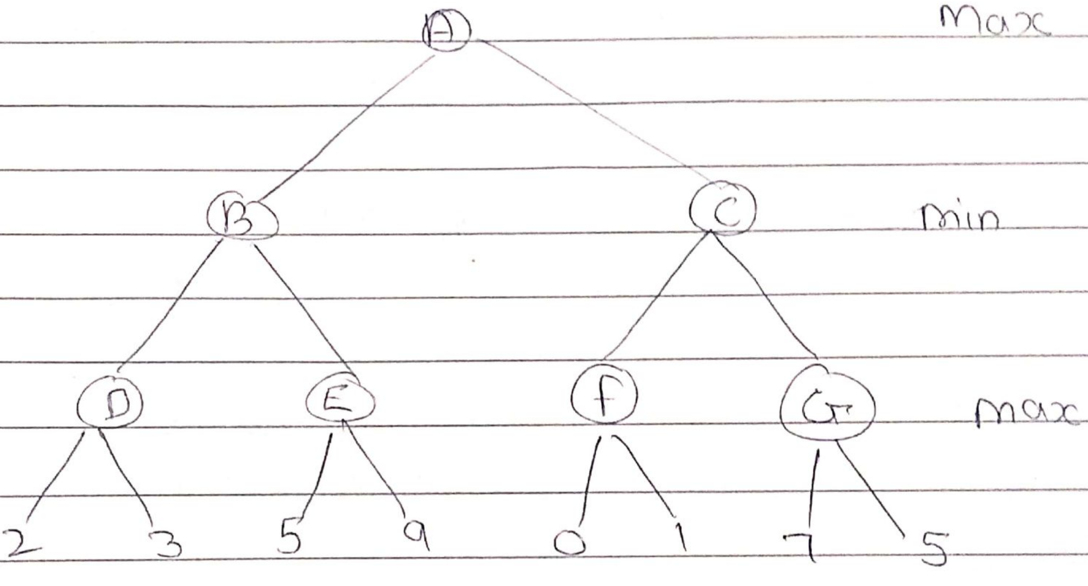
g) At Node 'E' [3, ∞)

α = max (3,0) = 3 = max (3,1)

β min (∞ 1) = 1

h) At 'A' [3, ∞]

Best value for A is max (3,1) = 3



2] a) At Node 'D' $[-\infty, \infty]$

$\alpha = \max(2, 3) = 3$ Node Val = 3

$\beta = \infty$

b) A Node 'B' $[-\infty, \infty]$

$\alpha = -\infty$

$\beta = \min(\infty, 3) = 3$

These values of B will be passed by next successor of B i.e E

c) At Node 'E' $[-\infty, 3]$

$\alpha = \max(-\infty, 5) = 5$ Node Val = 5

$\beta = 3$

$\alpha > \beta$ $\therefore$ right successor of E will be pruned & not traversed

d) At Node 'A' $[-\infty, \infty]$

$\alpha = \max(-\infty, 3) = 3$

$\beta = \infty$

These value of α is passed to next successor of A i.e, C

e) At Node 'E' $[3, \infty]$

- Same values will be passed to its next successor

f) At Node 'F' $[3, \infty]$

$\alpha = \max(3, 0) = 3 = \max(3, 1)$

$\beta = \infty$

Node value = 1