Expt 6:- Minimax Algorithm and Real World Problem (Tic Toe problem)

(*) Problem Formulation:-

(:) Here minimax algorithm is used to find the most optimal way of winning a tic tac toe problem.

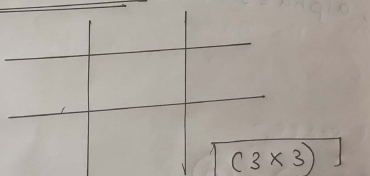
(*) Path cost:-

No. of permutations while solving the problem using state space.

(*) Operators:-

The no. of moves, signs and the state space are the operators.

(*) Initial State:-



(*) Algorithm:-

A description for the algorithm, assuming X is the "turn taking player" would look something like:-.

(:) If the game is over, return the score from X's perspective

c:) Otherwise get a list of new game states for every possible move.

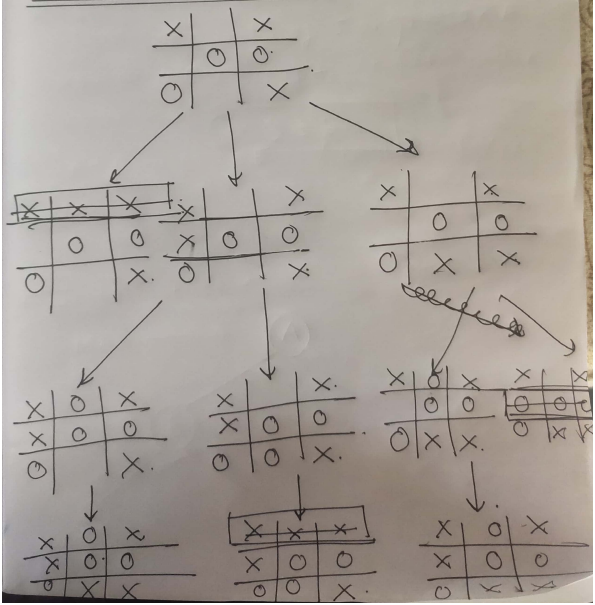
e:) Create a scores list.

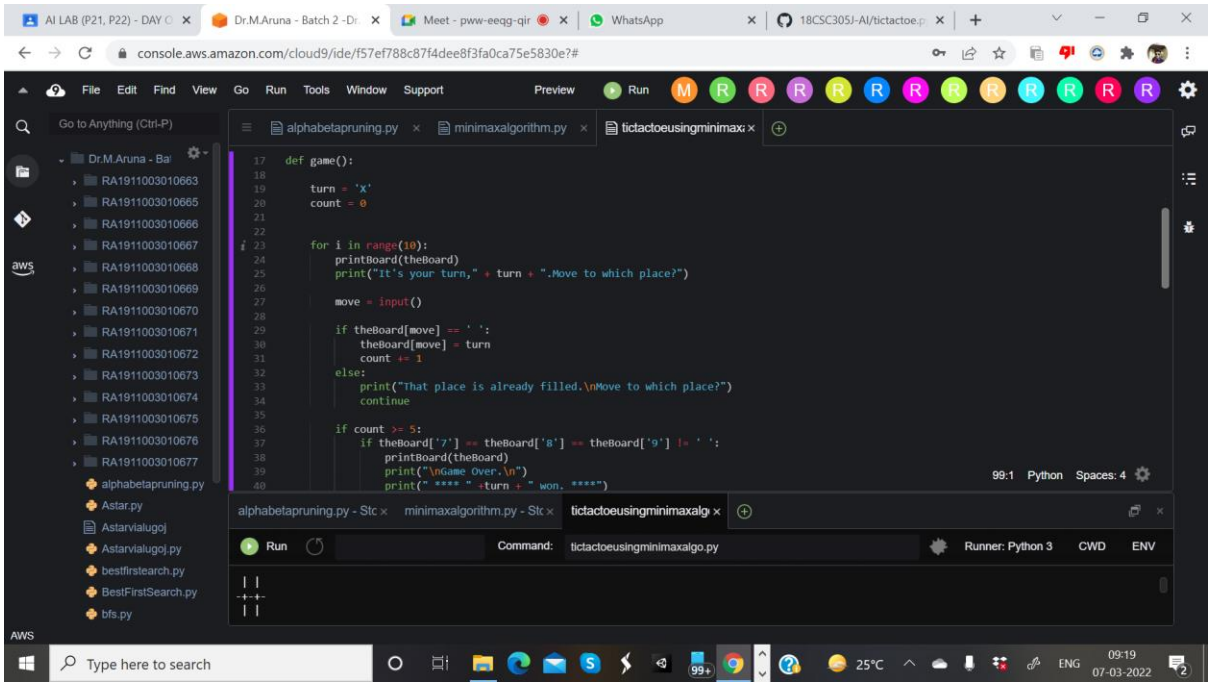
c:) For each of these states add the minimax result of that state to the scores list.

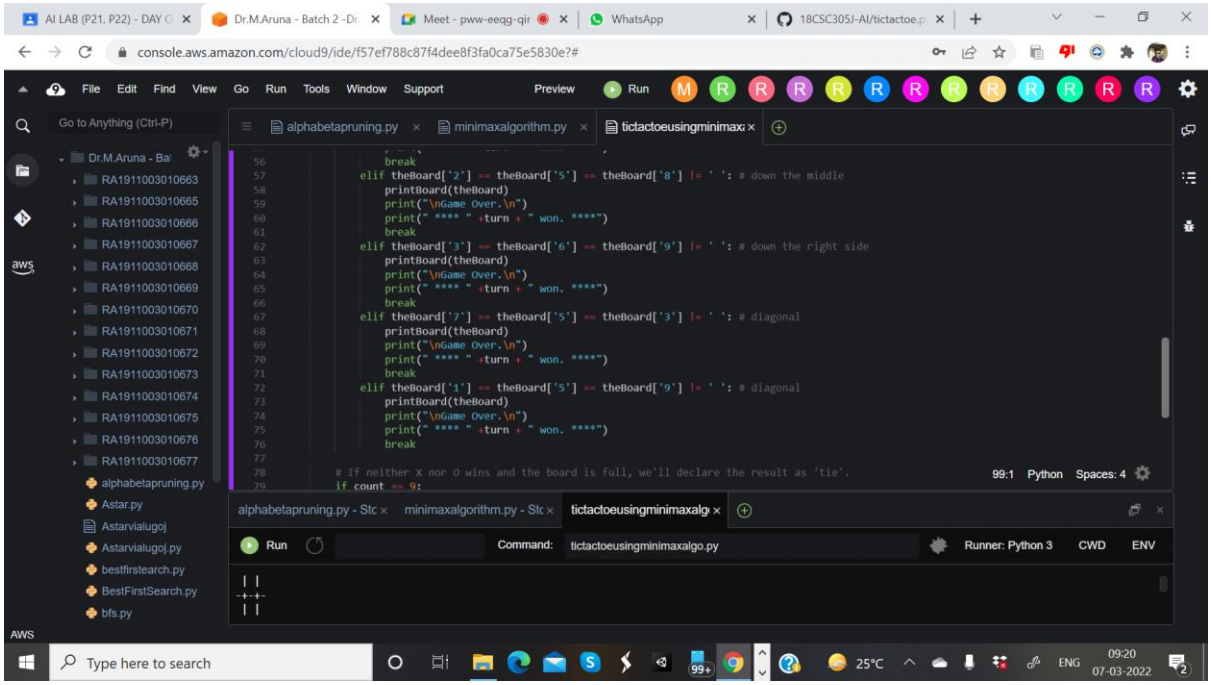
c:) If it's X's turn, return the maximum score from the scores list.

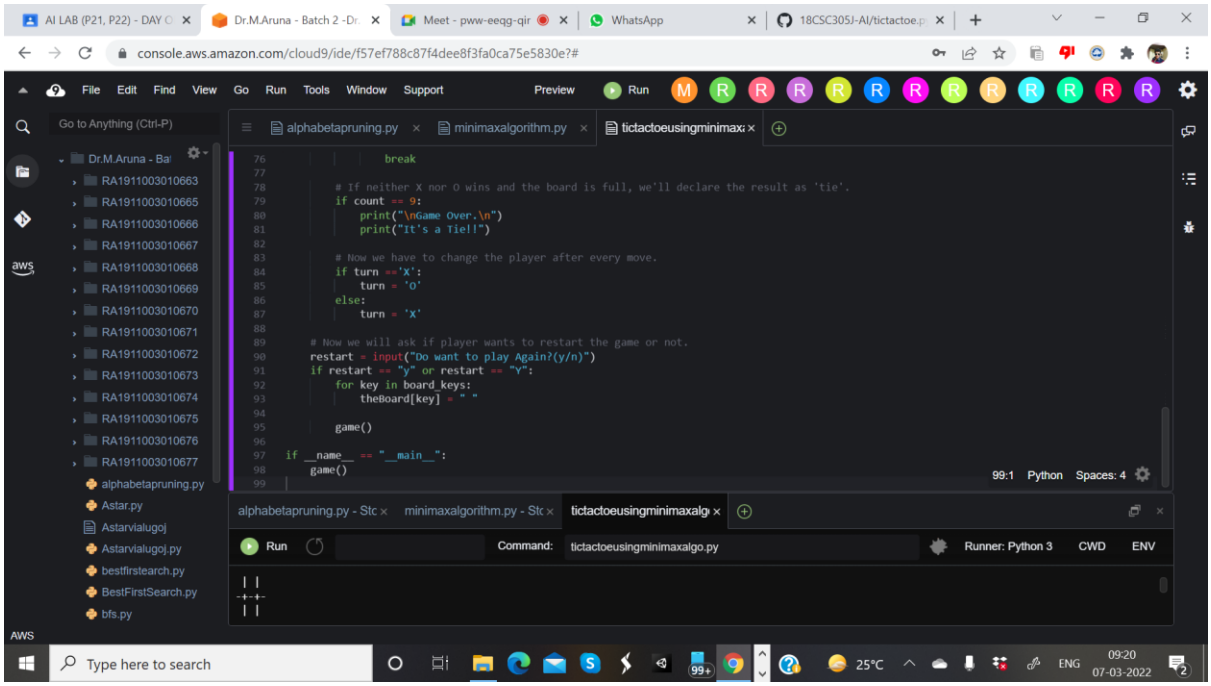
c:) If it's O's turn, return the maximum score from the scores list

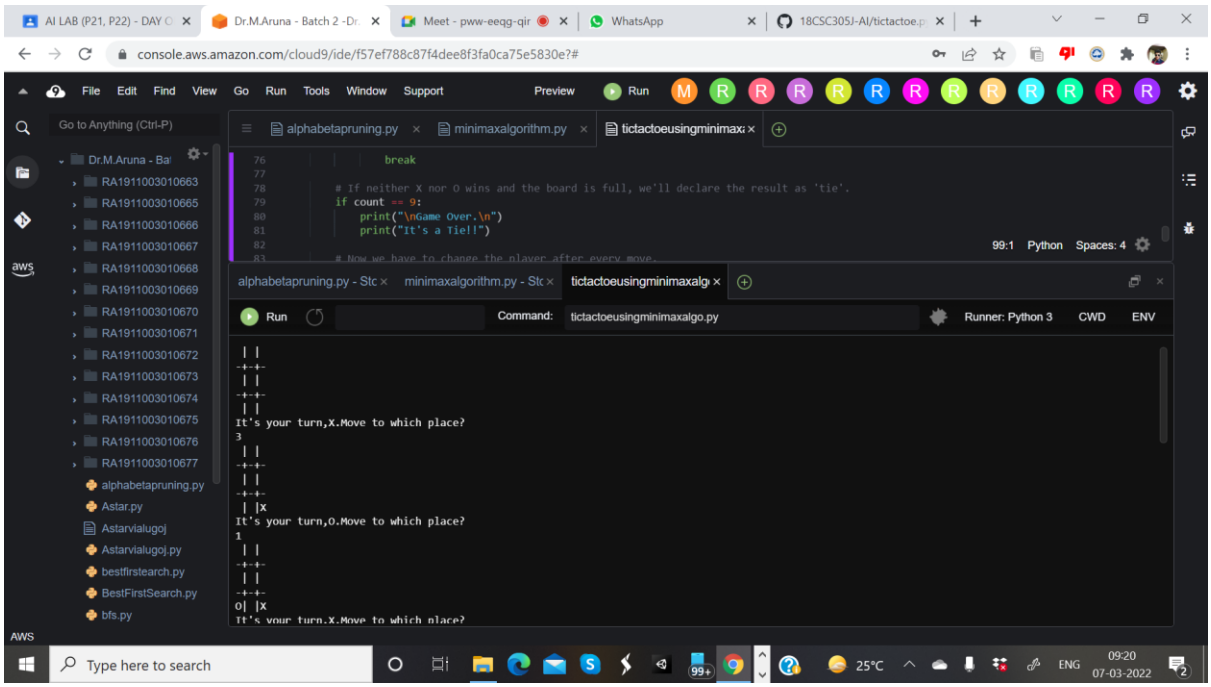
Goal ~~State~~ state:-

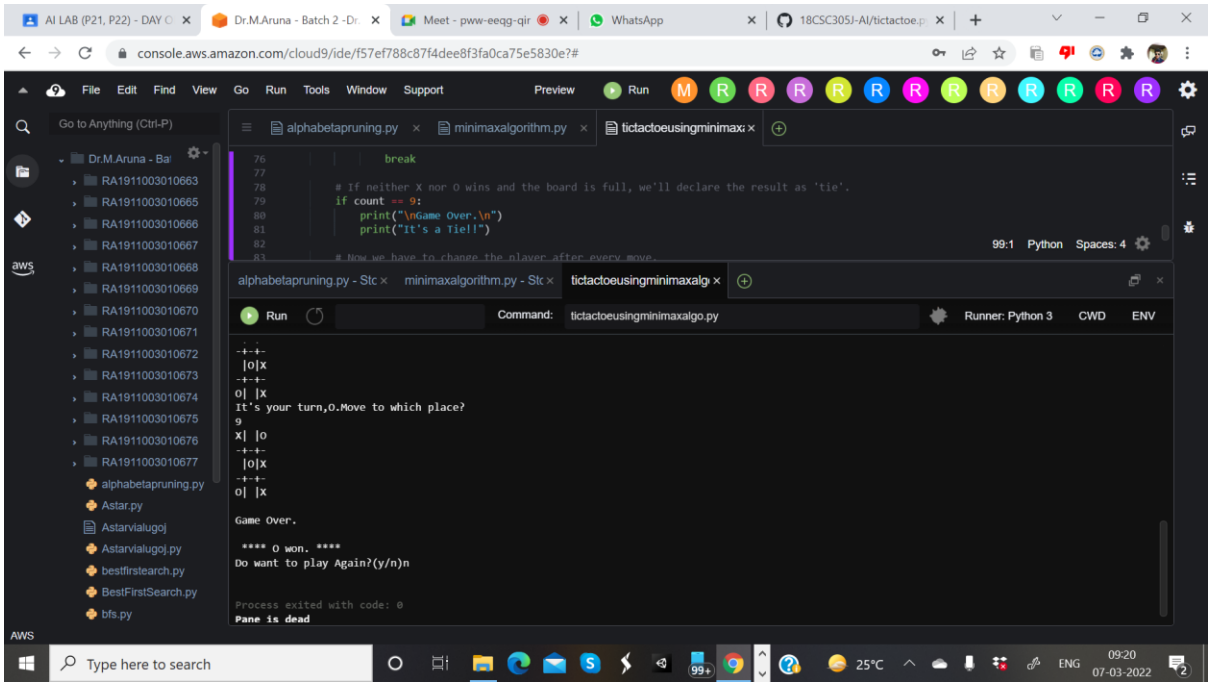


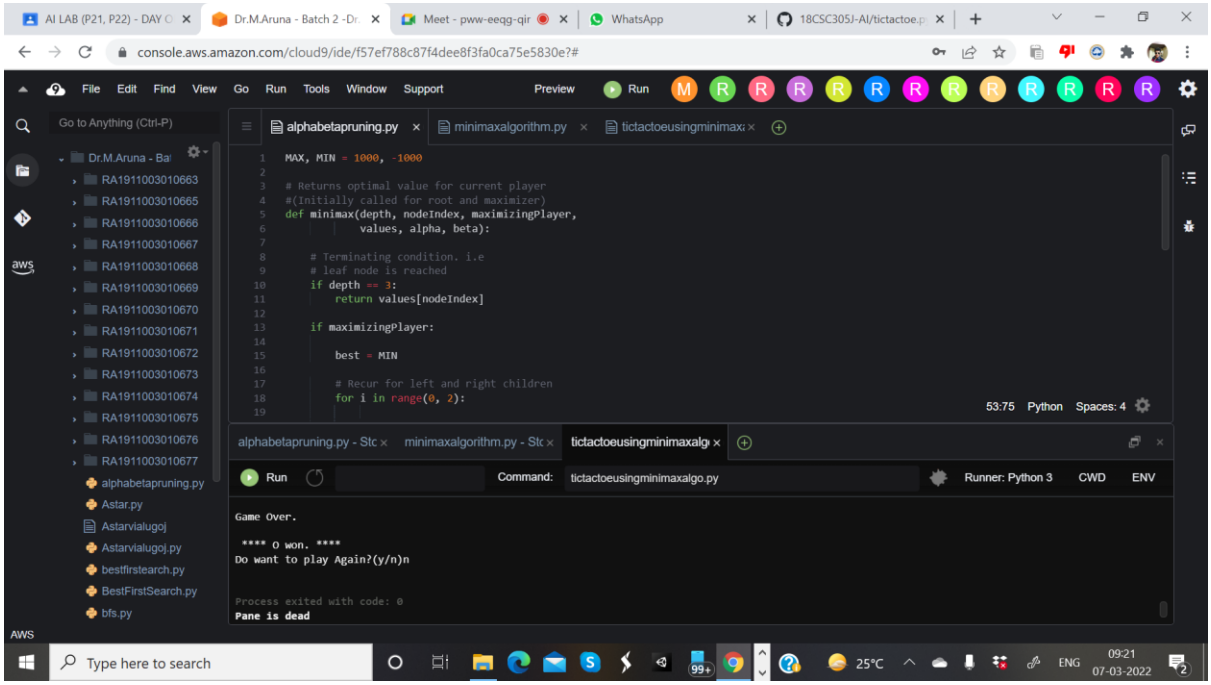

def game():
turn = 'X'
count = 0
for i in range(10):
printBoard(theBoard)
print("It's your turn," + turn + ".Move to which place?")
move = input()
if theBoard[move] == ' ':
theBoard[move] = turn
count += 1
else:
print("That place is already filled.\nMove to which place?")
continue
if count >= 5:
if theBoard['7'] == theBoard['8'] == theBoard['9'] != ' ':
printBoard(theBoard)
print("\nGame Over.\n")
print(" **** " +turn + " won. ****")
Run
Command: tictactoeusingminimaxalgo.py
Runner: Python 3
CWD
ENV
Type here to search

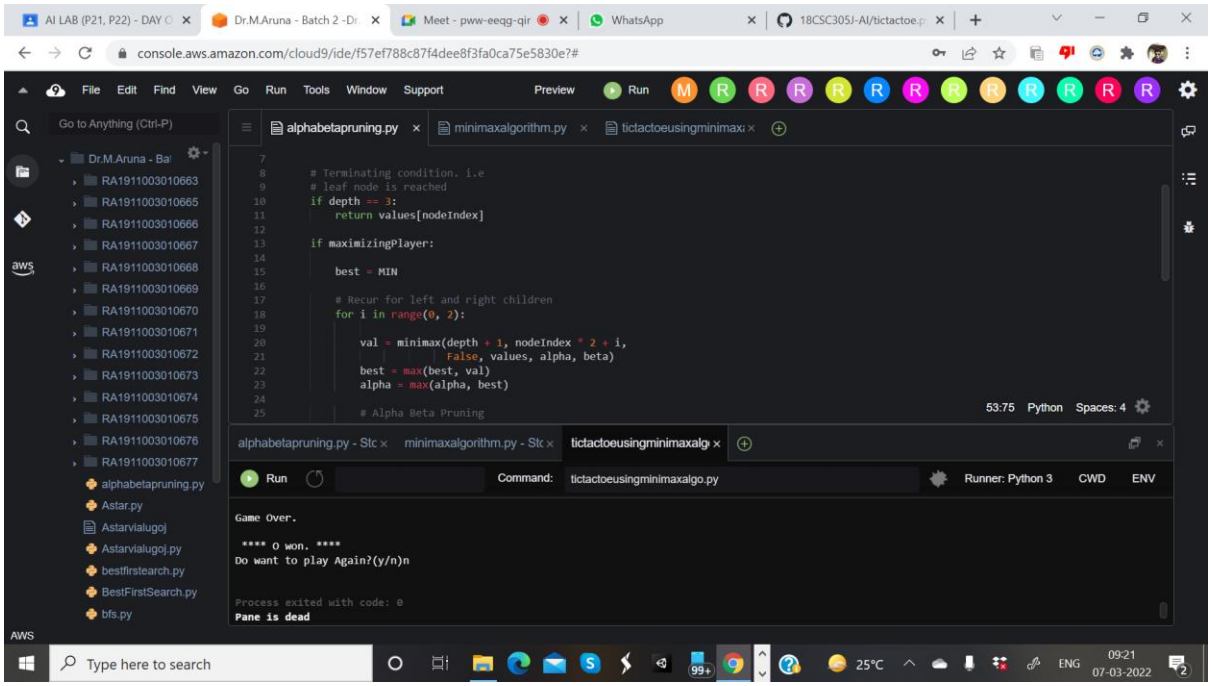

elif theBoard['2'] == theBoard['5'] == theBoard['8'] != ' ': # down the middle
printBoard(theBoard)
print("\nGame Over.\n")
print(" **** " +turn + " won. ****")
elif theBoard['3'] == theBoard['6'] == theBoard['9'] != ' ': # down the right side
printBoard(theBoard)
print("\nGame Over.\n")
print(" **** " +turn + " won. ****")
elif theBoard['7'] == theBoard['5'] == theBoard['3'] != ' ': # diagonal
printBoard(theBoard)
print("\nGame Over.\n")
print(" **** " +turn + " won. ****")
elif theBoard['1'] == theBoard['5'] == theBoard['9'] != ' ': # diagonal
printBoard(theBoard)
print("\nGame Over.\n")
print(" **** " +turn + " won. ****")
# If neither X nor O wins and the board is full, we'll declare the result as 'tie'.
if count == 9:
Run
Command: tictactoeusingminimaxalgo.py
Runner: Python 3
CWD
ENV
Type here to search

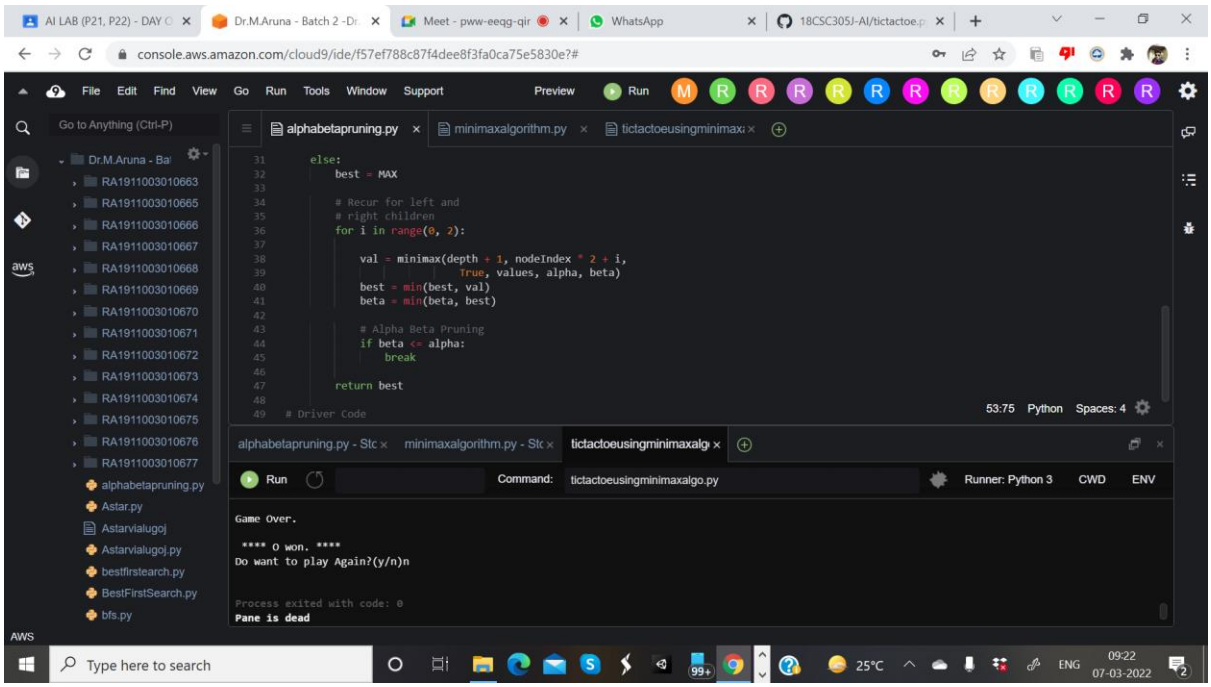

AI LAB (P21, P22) - DAY
Dr.M.Aruna - Batch 2 -Dr
Meet - pww-eeqg-qir
WhatsApp
18CSC305J-AI/tictactoe.p
console.aws.amazon.com/cloud9/ide/f57ef788c87f4dee8f3fa0ca75e5830e?#
File Edit Find View Go Run Tools Window Support Preview Run
Go to Anything (Ctrl-P)
alphabetapruning.py
minimaxalgorithm.py
tictactoeusingminimax
Dr.M.Aruna - Bat
RA1911003010663
RA1911003010665
RA1911003010666
RA1911003010667
RA1911003010668
RA1911003010669
RA1911003010670
RA1911003010671
RA1911003010672
RA1911003010673
RA1911003010674
RA1911003010675
RA1911003010676
RA1911003010677
alphabetapruning.py
Astar.py
Astarvialugoj
Astarvialugoj.py
bestfirstearch.py
BestFirstSearch.py
bfs.py
break
# If neither X nor O wins and the board is full, we'll declare the result as 'tie'.
if count == 9:
print("\nGame Over.\n")
print("It's a Tie!!")
# Now we have to change the player after every move.
if turn =='X':
turn = 'O'
else:
turn = 'X'
# Now we will ask if player wants to restart the game or not.
restart = input("Do want to play Again?(y/n)")
if restart == "y" or restart == "Y":
for key in board_keys:
theBoard[key] = " "
game()
if __name__ == "__main__":
game()
99:1 Python Spaces: 4
alphabetapruning.py - Stc
minimaxalgorithm.py - Stc
tictactoeusingminimaxalg
Run
Command:
tictactoeusingminimaxalgo.py
Runner: Python 3
CWD
ENV
| |
-+-+-
| |
AWS
Type here to search
25°C
ENG
09:20
07-03-2022

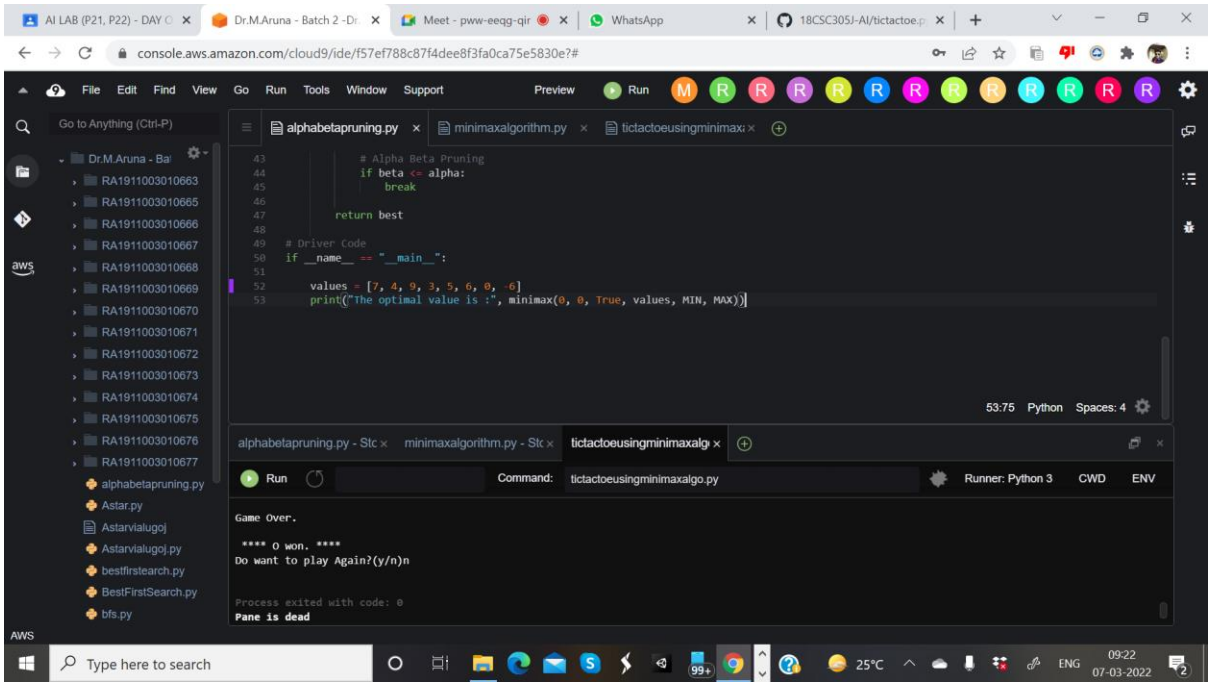

AI LAB (P21, P22) - DAY
Dr.M.Aruna - Batch 2 -Dr
Meet - pww-eeqg-qir
WhatsApp
18CSC305J-AI/tictactoe.p
console.aws.amazon.com/cloud9/ide/f57ef788c87f4dee8f3fa0ca75e5830e?#
File Edit Find View Go Run Tools Window Support Preview Run
Go to Anything (Ctrl-P)
alphabetapruning.py
minimaxalgorithm.py
tictactoeusingminimax
Dr.M.Aruna - Bat
RA1911003010663
RA1911003010665
RA1911003010666
RA1911003010667
RA1911003010668
RA1911003010669
RA1911003010670
RA1911003010671
RA1911003010672
RA1911003010673
RA1911003010674
RA1911003010675
RA1911003010676
RA1911003010677
alphabetapruning.py
Astar.py
Astarvialugoj
Astarvialugoj.py
bestfirstearch.py
BestFirstSearch.py
bfs.py
break
# If neither X nor O wins and the board is full, we'll declare the result as 'tie'.
if count == 9:
print("\nGame Over.\n")
print("It's a Tie!!")
99:1 Python Spaces: 4
alphabetapruning.py - Stc
minimaxalgorithm.py - Stc
tictactoeusingminimaxalg
Run
Command:
tictactoeusingminimaxalgo.py
Runner: Python 3
CWD
ENV
| |
-+-+-
| |
-+-+-
| |
It's your turn,X.Move to which place?
3
| |
-+-+-
| |
-+-+-
| |X
It's your turn,O.Move to which place?
1
| |
-+-+-
| |
-+-+-
O| |X
It's your turn,X.Move to which place?
AWS
Type here to search
25°C
ENG
09:20
07-03-2022


console.aws.amazon.com/cloud9/ide/f57ef788c87f4dee8f3fa0ca75e5830e?#
File Edit Find View Go Run Tools Window Support Preview Run
Go to Anything (Ctrl-P)
alphabetapruning.py
minimaxalgorithm.py
tictactoeusingminimax
break
# If neither X nor O wins and the board is full, we'll declare the result as 'tie'.
if count == 9:
print("\nGame Over.\n")
print("It's a Tie!!")
99:1 Python Spaces: 4
Run Command: tictactoeusingminimaxalgo.py Runner: Python 3 CWD ENV
-+-+-
|o|x
-+-+-
o| |x
It's your turn,O.Move to which place?
9
x| |o
-+-+-
|o|x
-+-+-
o| |x
Game Over.
**** O won. ****
Do want to play Again?(y/n)n
Process exited with code: 0
Pane is dead
Type here to search
25°C
ENG
09:20
07-03-2022


console.aws.amazon.com/cloud9/ide/f57ef788c87f4dee8f3fa0ca75e5830e?#
File Edit Find View Go Run Tools Window Support Preview Run
Go to Anything (Ctrl-P)
alphabetapruning.py
minimaxalgorithm.py
tictactoeusingminimax
MAX, MIN = 1000, -1000
Returns optimal value for current player
#(Initially called for root and maximizer)
def minimax(depth, nodeIndex, maximizingPlayer,
values, alpha, beta):
# Terminating condition. i.e
# leaf node is reached
if depth == 3:
return values[nodeIndex]
if maximizingPlayer:
best = MIN
# Recur for left and right children
for i in range(0, 2):
53:75 Python Spaces: 4
Run Command: tictactoeusingminimaxalgo.py Runner: Python 3 CWD ENV
Game Over.
**** O won. ****
Do want to play Again?(y/n)n
Process exited with code: 0
Pane is dead
Type here to search
25°C
ENG
09:21
07-03-2022


AI LAB (P21, P22) - DAY
Dr.M.Aruna - Batch 2 -Dr
Meet - pww-eeqg-qir
WhatsApp
18CSC305J-AI/tictactoe.p
console.aws.amazon.com/cloud9/ide/f57ef788c87f4dee8f3fa0ca75e5830e?#
File Edit Find View Go Run Tools Window Support Preview Run
Go to Anything (Ctrl-P)
alphabetapruning.py
minimaxalgorithm.py
tictactoeusingminimax
Dr.M.Aruna - Bat
RA1911003010663
RA1911003010665
RA1911003010666
RA1911003010667
RA1911003010668
RA1911003010669
RA1911003010670
RA1911003010671
RA1911003010672
RA1911003010673
RA1911003010674
RA1911003010675
RA1911003010676
RA1911003010677
alphabetapruning.py
Astar.py
Astarvialugoj
Astarvialugoj.py
bestfirstearch.py
BestFirstSearch.py
bfs.py
# Terminating condition. i.e
# leaf node is reached
if depth == 3:
return values[nodeIndex]
if maximizingPlayer:
best = MIN
# Recur for left and right children
for i in range(0, 2):
val = minimax(depth + 1, nodeIndex * 2 + i,
False, values, alpha, beta)
best = max(best, val)
alpha = max(alpha, best)
# Alpha Beta Pruning
53:75 Python Spaces: 4
alphabetapruning.py - Stc
minimaxalgorithm.py - Stc
tictactoeusingminimaxalg
Run
Command: tictactoeusingminimaxalgo.py
Runner: Python 3 CWD ENV
Game Over.
**** O won. ****
Do want to play Again?(y/n)n
Process exited with code: 0
Pane is dead
AWS
Type here to search
25°C
ENG
09:21
07-03-2022


AI LAB (P21, P22) - DAY
Dr.M.Aruna - Batch 2 -Dr
Meet - pww-eeqg-qir
WhatsApp
18CSC305J-AI/tictactoe.p
console.aws.amazon.com/cloud9/ide/f57ef788c87f4dee8f3fa0ca75e5830e?#
File Edit Find View Go Run Tools Window Support Preview Run
Go to Anything (Ctrl-P)
alphabetapruning.py
minimaxalgorithm.py
tictactoeusingminimax
Dr.M.Aruna - Bat
RA1911003010663
RA1911003010665
RA1911003010666
RA1911003010667
RA1911003010668
RA1911003010669
RA1911003010670
RA1911003010671
RA1911003010672
RA1911003010673
RA1911003010674
RA1911003010675
RA1911003010676
RA1911003010677
alphabetapruning.py
Astar.py
Astarvialugoj
Astarvialugoj.py
bestfirstearch.py
BestFirstSearch.py
bfs.py
else:
best = MAX
# Recur for left and
# right children
for i in range(0, 2):
val = minimax(depth + 1, nodeIndex * 2 + i,
True, values, alpha, beta)
best = min(best, val)
beta = min(beta, best)
# Alpha Beta Pruning
if beta <= alpha:
break
return best
Driver Code
53:75 Python Spaces: 4
alphabetapruning.py - Stc
minimaxalgorithm.py - Stc
tictactoeusingminimaxalg
Run
Command: tictactoeusingminimaxalgo.py
Runner: Python 3 CWD ENV
Game Over.
**** O won. ****
Do want to play Again?(y/n)n
Process exited with code: 0
Pane is dead
AWS
Type here to search
25°C
ENG
09:22
07-03-2022


AI LAB (P21, P22) - DAY
Dr.M.Aruna - Batch 2 -Dr
Meet - pww-eeqg-qir
WhatsApp
18CSC305J-AI/tictactoe.p
console.aws.amazon.com/cloud9/ide/f57ef788c87f4dee8f3fa0ca75e5830e?#
File Edit Find View Go Run Tools Window Support Preview Run
Go to Anything (Ctrl-P)
Dr.M.Aruna - Ba
RA1911003010663
RA1911003010665
RA1911003010666
RA1911003010667
RA1911003010668
RA1911003010669
RA1911003010670
RA1911003010671
RA1911003010672
RA1911003010673
RA1911003010674
RA1911003010675
RA1911003010676
RA1911003010677
alphabetapruning.py
Astar.py
Astarvialugoj
Astarvialugoj.py
bestfirstearch.py
BestFirstSearch.py
bfs.py
AWS
alphabetapruning.py
minimaxalgorithm.py
tictactoeusingminimax
# Alpha Beta Pruning
if beta <= alpha:
break
return best
# Driver Code
if __name__ == "__main__":
values = [7, 4, 9, 3, 5, 6, 0, -6]
print("The optimal value is :", minimax(0, 0, True, values, MIN, MAX))
53:75 Python Spaces: 4
alphabetapruning.py - Stc
minimaxalgorithm.py - Stc
tictactoeusingminimaxalgo
Run
Command: tictactoeusingminimaxalgo.py
Runner: Python 3 CWD ENV
Game Over.
**** O won. ****
Do want to play Again?(y/n)n
Process exited with code: 0
Pane is dead
Type here to search
25°C
ENG
09:22
07-03-2022