**CSE422 Lab Assignment03**

"""

@author: Simon Biswas

"""

import random

import math

MAX = math.inf

MIN = -math.inf

p1 = 0

def minimax(depth, nodeIndex, maximizingPlayer,values, alpha, beta):

global p1

if depth == 0:

return values[nodeIndex]

if maximizingPlayer:

for w in range(0,numberOfTurns): # working keeping in mind about numberOfTurns

maxEval = MIN

for i in range(0, numberOfBranchs):

val = minimax(depth - 1, nodeIndex + 1, False, values, alpha, beta)

maxEval = max(maxEval, val)

alpha = max(alpha, maxEval)

# Checking and counting alpha beta pruning

if beta <= alpha:

p1 = p1 + 1

break

return maxEval

else:

for w in range(0, numberOfTurns): # working keeping in mind about numberOfTurns

minEval = MAX

for i in range(0, numberOfBranchs):

val = minimax(depth - 1, nodeIndex + 1, True, values, alpha, beta)

minEval = min(minEval, val)

beta = min(minEval, minEval)

# Alpha Beta Pruning

if beta <= alpha:

break

return minEval

# main function

if \_\_name\_\_ == "\_\_main\_\_":

depth = int(3) # we are taking 3 as we will terminate when depth==0 means we completed reaching top

inputFile = open("Lab03\_Input.txt", "r")

m = inputFile.readlines()

numberOfTurns = int(m[0])

numberOfBranchs = int(m[1])

m1 = [int(n) for n in m[2].split()]

minRange = m1[0]

maxRange = m1[1]

terminalStates = int(math.pow(numberOfBranchs, depth - 1))

p = list()

for i in range(terminalStates + 1):

p.append(int(random.randint(minRange, maxRange)))

values = p # random values

print("Depth:", depth - 1)

print("Branch:", numberOfBranchs)

print("Terminal States:", terminalStates)

minimax(depth, 0, False, values, MIN, MAX)

print("Comparison (Before alpha-beta pruning) :", terminalStates)

print("Comparison (After alpha-beta pruning) :", terminalStates - p1)

**Lab03\_Input**

1

3

1 20