import random

def minimax(position,depth,alpa, beta,maximizingplayer):

global comparison

if depth == 0:

return leaf\_values[position]

if maximizingplayer:

maxEval = maximum

for i in range(0,branch):

child = position\*branch+i

eval = minimax(child,depth-1,alpa,beta,False)

maxEval = max(maxEval,eval)

alpa = max(alpa,eval)

if beta<=alpa:

comparison += 1

break

return maxEval

else:

minEval = minimum

for i in range(0,branch):

child = position \* branch + i

eval = minimax(child, depth-1,alpa, beta, True)

minEval = min(minEval, eval)

beta = min(beta, eval)

if beta <= alpa:

comparison += 1

break

return minEval

minimum = 99999999999999999

maximum = -99999999999999999

count = 0

turns = int(input("Pleae enter the number of turns for Riyad: "))

depth = 2\*turns

branch=int(input("Please enter the branching number: "))

num\_of\_leaf\_nodes = int(branch\*\*depth)

lowest\_range = int(input("Please enter the lowest range value: "))

highest\_range = int(input("Please enter the highest range value: "))

leaf\_values=[]

for i in range(0,num\_of\_leaf\_nodes):

i = random.randint(lowest\_range,highest\_range)

leaf\_values.append(i)

#leaf\_values = [3, 12, 8, 2, 4, 6, 13, 5, 2]

#print(leaf\_values)

comparison = 0

print("Depth:",depth)

print("Branch:",branch)

print("Terminal States (Leaf Nodes): ",num\_of\_leaf\_nodes)

print("Maximum amount:", minimax(0,depth,maximum,minimum,True))

print("Comparisons:",num\_of\_leaf\_nodes)

print("comparisons:",num\_of\_leaf\_nodes-comparison)