# -\*- coding: utf-8 -\*-

"""8\_19301051\_Kazi Md. Al-Wakil\_CSE422\_Lab 3.ipynb

Automatically generated by Colaboratory.

Original file is located at

https://colab.research.google.com/drive/1qIfyWXNYmU5DXu039PbjkWHg6TYuMvSC

"""

#CSE422 Lab 3 (Alpha Beta Pruning- (Winner finding problem))

#Kazi Md. Al-Wakil

#19301051

import math as ongko

import random as rand

#main algo

def alifBataPrun(randList, dapthTrii, alif, bata, boolVar, travellingNod, branchinFactos):

if dapthTrii == 0:

return randList[travellingNod]

if boolVar!=False:

varAlif = alif

for itor in range(branchinFactos):

poinAchivedByOptPrim = alifBataPrun(randList, dapthTrii-1, alif, bata, False, (travellingNod\*branchinFactos)+itor, branchinFactos)

varAlif = max(poinAchivedByOptPrim, alif)

alif = max(varAlif, alif)

if alif>=bata:

break

return varAlif

else:

varBata = bata

for itor in range(branchinFactos):

poinAchivedByOptPrim = alifBataPrun(randList, dapthTrii-1, alif, bata, True, (travellingNod\*branchinFactos)+itor, branchinFactos)

varBata = min(poinAchivedByOptPrim, bata)

bata = min(varBata, bata)

if alif>=bata:

break

return varBata

#initializing all the values

def inputModificationAndFormatting(bracuVarsityId):

#converting all 0 to 8

bracuVarsityId = bracuVarsityId.replace("0","8")

#5th digit -> least poin the optim prm can achv frm the game

lowestLimitRange = int(bracuVarsityId[4])

#storing last 2 digit(in reverse) of id

pointsToWinByOptimus = int(bracuVarsityId[-1:-3:-1])

# highst poin the optim prm can achv frm the game

highestLimitRange = float(pointsToWinByOptimus \* 1.5)

if highestLimitRange.is\_integer() == False:

highestLimitRange = highestLimitRange + 0.5 #taking upper value of floating number

highestLimitRange = int(highestLimitRange)

#amount of shuffles or total Leaf nodes, 4th digit of student ID

totalShuff = int(bracuVarsityId[3])

#brnchin factos and dapth of trii

branchinFactos = 2

dapthTrii = 3

return lowestLimitRange, pointsToWinByOptimus, highestLimitRange, totalShuff, branchinFactos, dapthTrii

#taking inputs

bracuVarsityId = input()

#bracuVarsityId = "19301051"

#initializing all the values

lowestLimitRange, pointsToWinByOptimus, highestLimitRange, totalShuff, branchinFactos, dapthTrii = inputModificationAndFormatting(bracuVarsityId)

#generating random numbers from range (lest, highst)

randList = rand.sample(range(lowestLimitRange, highestLimitRange+1),totalShuff)

#alif bata initializing

alif = (ongko.inf)\*-1

bata = (ongko.inf)

#travelling Nod var initializing

travellingNod = 0 #this will traverse all the items in the list (randomly gen list)

print("Generated 8 random points between the minimum and maximum point ")

print("limits: ", randList)

print("Total points to win: ",pointsToWinByOptimus)

poinAchivedByOptPrim = alifBataPrun(randList, dapthTrii, alif, bata, True, travellingNod, branchinFactos)

print("Achieved point by applying alpha-beta pruning: ",poinAchivedByOptPrim)

if poinAchivedByOptPrim>=pointsToWinByOptimus:

print("The winner is Optimus Prime")

else:

print("The winner is Megatron")

#task 2

#shuffelin the randomly generated list and findin out the points won by optimus prime, storing them in a list

print("\nAfter the shuffle:")

pointsWonAfterEachShuffleList = []

for itor in range(totalShuff):

rand.shuffle(randList) #shuffleing

poinAchivedByOptPrim = alifBataPrun(randList, dapthTrii, alif, bata, True, travellingNod, branchinFactos) #after each shuffleing, calling the alif bata pruning algo

pointsWonAfterEachShuffleList.append(poinAchivedByOptPrim) #storing the points won in a list

counterVarJKotoBarJitse = 0

#counter to know and to see how many times mighty opt prim won

for itor in pointsWonAfterEachShuffleList:

if itor >= pointsToWinByOptimus:

counterVarJKotoBarJitse += 1

print("List of all points values from each shuffle:: ", pointsWonAfterEachShuffleList)

print("The maximum value of all shuffles: ", max(pointsWonAfterEachShuffleList))

print(f"Won {counterVarJKotoBarJitse} times out of {totalShuff} number of shuffles")