import random

class Node:

count=1

comparison=0

def \_\_init\_\_(self,prev=None,value=None,modifier=1):

self.count=Node.count

Node.count+=1

self.next=[]

self.value=value

self.modifier=modifier

self.prev=prev

def \_\_repr\_\_(self):

return str(self.count)

def tree(root, set, depth, choice, modifier=-1):

set\_size=choice\*\*(depth-1)

for a in range(choice):

#print(depth, set, modifier)

if depth==1:

new\_node=Node(prev=root, value=set[a],modifier=modifier)

root.next.append(new\_node)

print(depth,new\_node.count,new\_node.value,new\_node.next,new\_node.prev)

else:

new\_node=Node(prev=root,modifier=modifier)

root.next.append(new\_node)

tree(new\_node, set[a\*set\_size:(a+1)\*set\_size], depth-1, choice, modifier\*-1)

def minimax(root,alpha=-float("inf"),beta=float("inf")):

if len(root.next)==0:

Node.comparison+=1

return root.value

else:

count=0

while count<len(root.next) and alpha<beta:

val=minimax(root.next[count],alpha,beta)

#print(val, alpha, beta)

if root.modifier==1:

if val>alpha:

alpha=val

else:

if val<beta:

beta=val

count+=1

if root.modifier==1:

return alpha

else:

return beta

def printer(root,depth=0):

print(depth, root.value, root.next)

if len(root.next)==0:

print(depth, root.count, root.value, root.next,root.prev)

for a in root.next:

printer(a,depth+1)

def main():

id=input("Enter your student id: ")

att\_hp\_range=input("Minimum and Maximum value for the range of negative HP: ")

turn\_count=int(id[0])

def\_hp=int(id[-2:][::-1])

#print(def\_hp)

bullet\_count=int(id[2])

att\_hp\_range=att\_hp\_range.rstrip().split()

att\_hp\_range=[int(a) for a in att\_hp\_range]

#print(att\_hp\_range)

depth=turn\_count\*2

print("1. Depth and Branches ratio is",str(depth)+":"+str(bullet\_count))

print("2. Terminal States (leaf node values) are ",end="")

state\_array=[random.randint(att\_hp\_range[0],att\_hp\_range[1]) for a in range(bullet\_count\*\*depth)]

#state\_array = [19,22,9,2,26,16,16,27,16]

#state\_array = [18,13,5,12,10,5,13,7,17,8,6,8,5,11,13,18]

for a in state\_array[:-1]:

print(str(a)+",",end="")

print(state\_array[-1], end=".\n")

#print(state\_array)

root=Node()

tree(root, state\_array, depth, bullet\_count)

#printer(root)

damage\_dealt=minimax(root)

comparison\_count=Node.comparison

print("3. Left life(HP) of the defender after maximum damage caused by the attacker is",def\_hp-damage\_dealt)

print("4. After Alpha-Beta Pruning Leaf Node Comparisons",comparison\_count)

#17301106

#1 30

#

#2:3

#44

#7

#20201003

#5 20

#

#4:2

#22

#13

#All above values are proper tested results

main()