**import random**

**import math**

***# import numpy as np***

**inf=math.inf**

**def alphabetaPruning(pos,d,maximizingPLayer,p,ch,depthAr,ev,a,b):**

**if d==0:**

**global alphabetaComparison**

**alphabetaComparison+=1**

**return ev[pos]**

**if maximizingPLayer:**

**maxEval=-inf**

**tempChild=ch[pos]**

**for i in range(len(tempChild)):**

**ev[i]=alphabetaPruning(tempChild[i],d-1,False,p,ch,depthAr,ev,a,b)**

**maxEval=max(maxEval,ev[i])**

**a=max(a,ev[i])**

**if b<=a:**

**break**

**return maxEval**

**else:**

**minEval=inf**

**tempchild=ch[pos]**

**for i in range(len(tempchild)):**

**ev[i]=alphabetaPruning(tempchild[i],d-1,True,p,ch,depthAr,ev,a,b)**

**minEval=min(minEval,ev[i])**

**b=min(b,ev[i])**

**if b<=a:**

**break**

**return minEval**

**def minimax(pos, d,maximizingPLayer,p,ch,depthAr,ev):**

**if d==0:**

**return ev[pos]**

**if maximizingPLayer:**

**maxEval=-inf**

**tempChild=ch[pos]**

**for i in range(len(tempChild)):**

**ev[i]=minimax(tempChild[i],d-1,False,p,ch,depthAr,ev)**

**maxEval=max(maxEval,ev[i])**

**return maxEval**

**else:**

**minEval=inf**

**tempchild=ch[pos]**

**for i in range(len(tempchild)):**

**ev[i]=minimax(tempchild[i],d-1,True,p,ch,depthAr,ev)**

**minEval=min(minEval,ev[i])**

**return minEval**

**if \_\_name\_\_ == '\_\_main\_\_':**

**f=open("tester.txt")**

**turn=int(f.readline())**

**depth=turn\*2**

**branch=int(f.readline())**

**minRange,maxRange=map(int, f.readline().split())**

***# print(maxRange)***

**leafNode= branch\*\*depth**

***# print("leafNode= ",leafNode)***

**totalNode = 0**

**for i in range(depth + 1):**

**totalNode = totalNode + (branch \*\* i)**

***# print("Total Node= ", totalNode)***

**parent=[None for i in range(totalNode)]**

***# print("parent before :",parent)***

**for i in range(totalNode-leafNode):**

**counter=0**

**for j in range(branch):**

**parent[i\*branch+branch-counter]=i *#making the parent array/main array***

**counter+=1**

***# print("parent after: ",parent)***

**depthCnt=0**

**maxDepth=depth**

**depArray = [None for i in range(totalNode)]**

**for i in range(depth+1): *#making the depth array***

**for j in range(branch\*\*i):**

**depArray[depthCnt]=maxDepth**

**depthCnt+=1**

**maxDepth -= 1**

***# print("Depth array= ",depArray)***

**children=[[] for i in range(totalNode-leafNode)]**

**for i in range(len(children)):**

**for j in range(len(parent)):**

**if parent[j]==i:**

**children[i].append(j)**

***# print("Children array :",children)***

**eValArray=[None for i in range(totalNode)]**

**for i in range(len(eValArray)):**

**if depArray[i]==0:**

**eValArray[i]=random.randint(minRange,maxRange) *#np.random.randint(minRange,maxRange)***

**print("Depth: ",depth)**

**print("Branch: ",branch)**

**print("Terminal states: ",leafNode)**

**alphabetaComparison = 0**

**m=minimax(0,depth,True,parent,children,depArray,eValArray)**

***# print("After minmax: ",m)***

**n=alphabetaPruning(0,depth,True,parent,children,depArray,eValArray,-inf,inf)**

***# print("After pruning: ",n)***

**print("Max amount: ",n)**

**print("Comparisons after minimax: ",leafNode)**

**print("Comparisons after apruning: ",alphabetaComparison)**