Lab 3 Assignments

Problem No 1:

My friend came with

1. Karatsuba algorithm performs the multiplication on binary numbers more efficiently than decimal numbers, but the problem is we cannot visualize Karatsuba’s algorithm step by step,
   * 1000000000 base 2 (10 bits)
   * 1024 base 10 (4 digits)
   * 400 base 16 (3 Hexa)

c): In integer multiplication, Karatsuba’s algorithms fail on larger numbers of integers as we cannot divide such large numbers into 3 parts. both given numbers are equal but in different bases, so lengths are different. In the Karatsuba algorithm, we cannot work perfectly with a lengthy number.

Problem No 2:

Part a):

Input=[[1,4],[2,5],[7,9],[9,10],[6,10]]

def friendSlower(Input):

AllPairs=[]

for i in range(len(Input)):

for j in range(i,len(Input)):

if I ! = j:

Pair = []

if Input[j][0]< =Input[i][0] &&Input[j][1]>=Input[i][0] || Input[i][1]<=Input[j][1] && Input[i][1]>=Input[j][0]:

Pair[i] =(i+1)

Pair[j] =(j+1)

All Pairs[i]= (Pair)

return tuple(map(tuple,AllPairs))

pair=friend Slower(Input)

print(pair)

This algorithm work by comparing the index of the array with the indexes of arrays. such as making a 2d array comparing the 2nd element with the 1st.if it finds the 2nd element of the 1st array equal to either of the starting or ending index and equal to the same index it will store the value of the value being compared or to the value being compared with. As it is asked in the manual to return an answer as a tuple so outside the loops the 2D array is converted into the Tuple so we map tuple with list

Problem No 3:

Part a)

import random

def getPopulation(n):

array=[]

for i:0 to length n:

array[i]= bool(random.randint(0, 1))

random.shuffle(array)

return array

def TruthFulToadsA(population):

lst=[]

for i :0 to length of population:

if(population[i]==1):

population[i]="Trustworthy"

else:

population[i]="Tricky"

for i:0 to length of population:

if ( population[i] == "Trustworthy" ):

{lst.append(i)}

return lst

def TruthFulToadsB(population):

lst1=[]

arr=[]

for i :0 to length (population)//2):

lst1[i]=random.randint(1,1)

for j :0 to length population)//2:

lst1[j]= random.randint(0,1)

random.shuffle(lst1)

for m:0 to length population:

{

if(population[m]==1):

{arr[m]=lst1[i]}

else:

ran=random.random()

if(ran<0.5):

{

arr.[m]=1

}

}else

{

arr.[m]=0

}

}

lst2=[]

for m:0 to length arr

{

if(arr[m] == 1)

{

lst2[m]

}

}

return lst2

num=int(input("Enter population : "))

population=getPopulation(num)

pop=TruthFulToadsA(population)

pop1=TruthFulToadsB(population)

print('Tode A say {} speak true'.format(pop))

print('Tode B say {} speak true'.format(pop1))

Part d and e )

I have not implemented recursive part of part a and b so I can not provide solution of these answers.

Part g)

to find all the trustworthy toads in the Array we use the following procedures:

We first use a for loop iteration to find the toads, After comparison there are many or tricky or trustworthy todes in our Array.in the next step we compare these array with original arrays Then we use a condition to find if the toad is trustworthy or not if it is then the index of that toad is append to another array and if not then the index is ignored and trustworthy toads index .