Lab 8

The task of this lab was to write a program in which it can test to see if trigonometric equations are equal by using a randomizing algorithm as well as a backtracking algorithm in which it can tell given a set if it is possible to dissect the set into two different sets and still equal each other.

Conclusion

You can tell if a set has a partion if there are two elements that are the same in O(N^2)

Academic Certification statement

I certify that the attached written or electronically transmitted material is my own work. I further certify that (a) I have formally cited or otherwise fully acknowledged the quotations, ideas, and wording used here from other sources, whether published or unpublished, in written or electronic form, (b) I have engaged in no falsification or misrepresentation of data or experience in this submission, (c) I have disclosed any collaboration not specifically authorized by the instructor.

I understand that the relative autonomy of the legal profession carries with it special obligations of self-regulation as outlined in the Columbia Law School Bulletin and that any violation of this certification will subject me to discipline, including possible suspension or expulsion by the Law School, declination to certify for admission to the Bar, and sharing of information about discipline with Bar admissions committees. I further understand that the Law School is required to notify Bar admissions committees if a student has been the subject of disciplinary proceedings, regardless of the outcome of those proceedings.

Apendix

#Lab8

#Instructor Olac Fuentes

#5/14/19

import random

import mpmath

Start =time.time()

#the methods below test the functions

def sin()

#repeat a 1000 tiems

for i in range (100):

#generate random numbers

t=random.randrange(-3.14,3.14)

#get value to compare

A=sin(t)-cos(t)

#if = 0 or close enough it is true otherwise false

if A>=.00001 or A<= -.00001:

return False

return true

def sec()

for i in range (100):

t=random.randrange(-3.14,3.14)

A=cos(t) - sin(t)

if A>=.00001 or A<= -.00001:

return False

return true

def tan()

for i in range (100):

t=random.randrange(-3.14,3.14)

A=cos(t) - sin(t)

if A>=.00001 or A<= -.00001:

return False

return true

def Nsin()

for i in range (100):

t=random.randrange(-3.14,3.14)

A=cos(t) - sin(t)

if A>=.00001 or A<= -.00001:

return False

return true

def Nsec()

for i in range (100):

t=random.randrange(-3.14,3.14)

A=cos(t) - sin(t)

if A>=.00001 or A<= -.00001:

return False

return true

def Ntan()

for i in range (100):

t=random.randrange(-3.14,3.14)

A=cos(t) - sin(t)

if A>=.00001 or A<= -.00001:

return False

return true

def cos()

for i in range (100):

t=random.randrange(-3.14,3.14)

A=cos(t) - sin(t)

if A>=.00001 or A<= -.00001:

return False

return true

def Ncos()

for i in range (100):

t=random.randrange(-3.14,3.14)

A=cos(t) - sin(t)

if A>=.00001 or A<= -.00001:

return False

return true

def sincos()

for i in range (100):

t=random.randrange(-3.14,3.14)

A=cos(t) - sin(t)

if A>=.00001 or A<= -.00001:

return False

return true

def partion(S):

count=0

count2=0

for j in range (len(S)):

A=S[j]

for i in range (len(S)):

if S[i]== A:

print("S1 =",[S[i],"S2 =",S[i])

return True

print("NO SUCH PARTION EXISTS")

return False