PAOLA TERRAZAS

CS2302

DISCOVER TRIG IDENTITIES USING RANDOMIZATION

DR. FUENTES

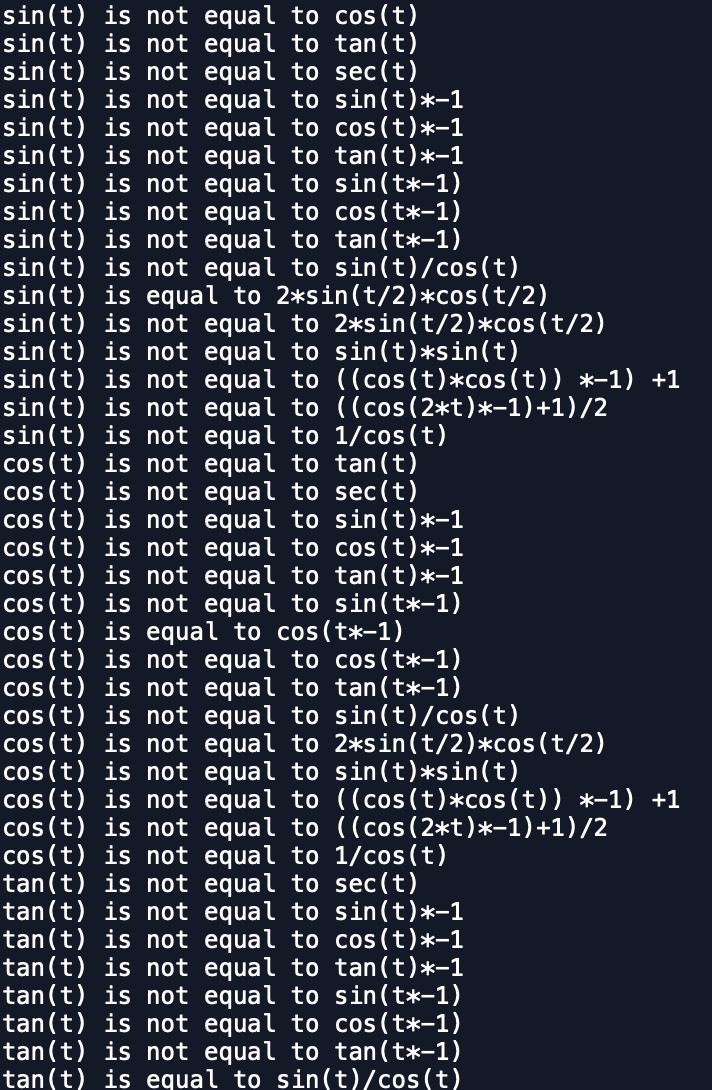
TA – ANINDITA NATH

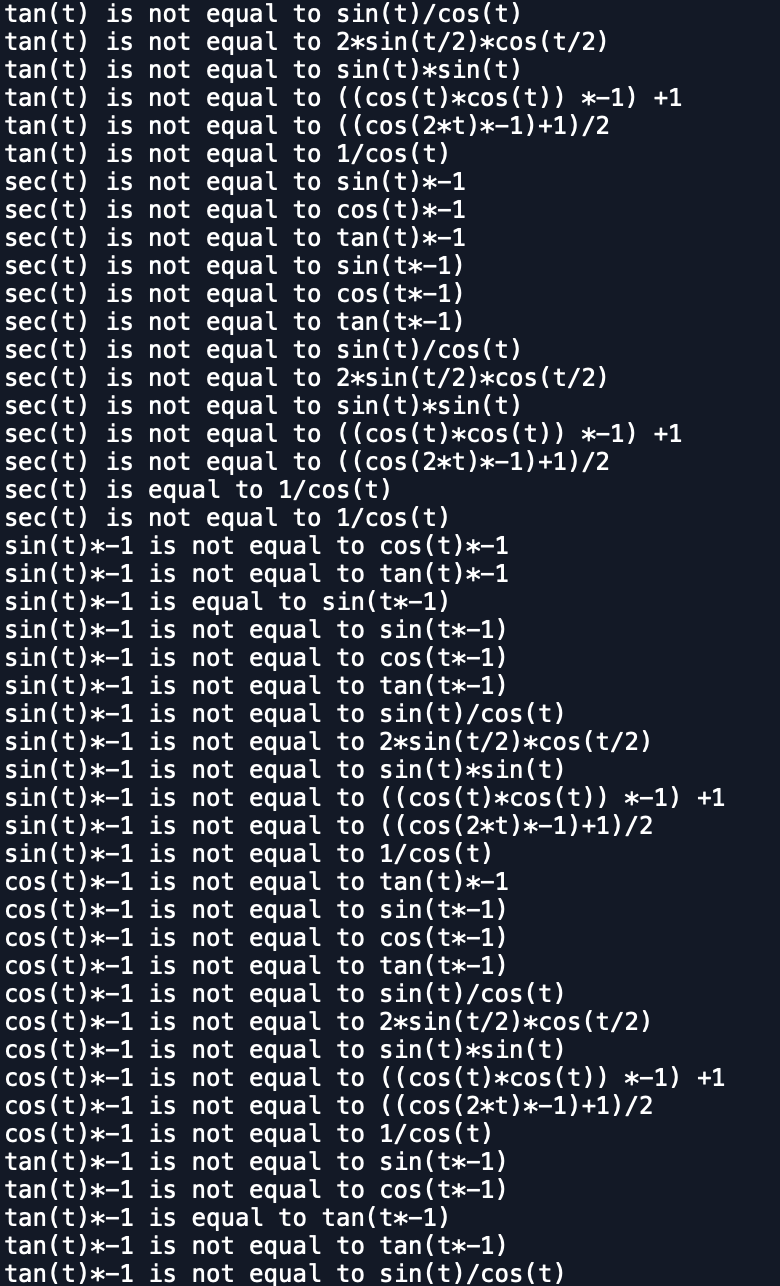
5.10.2019

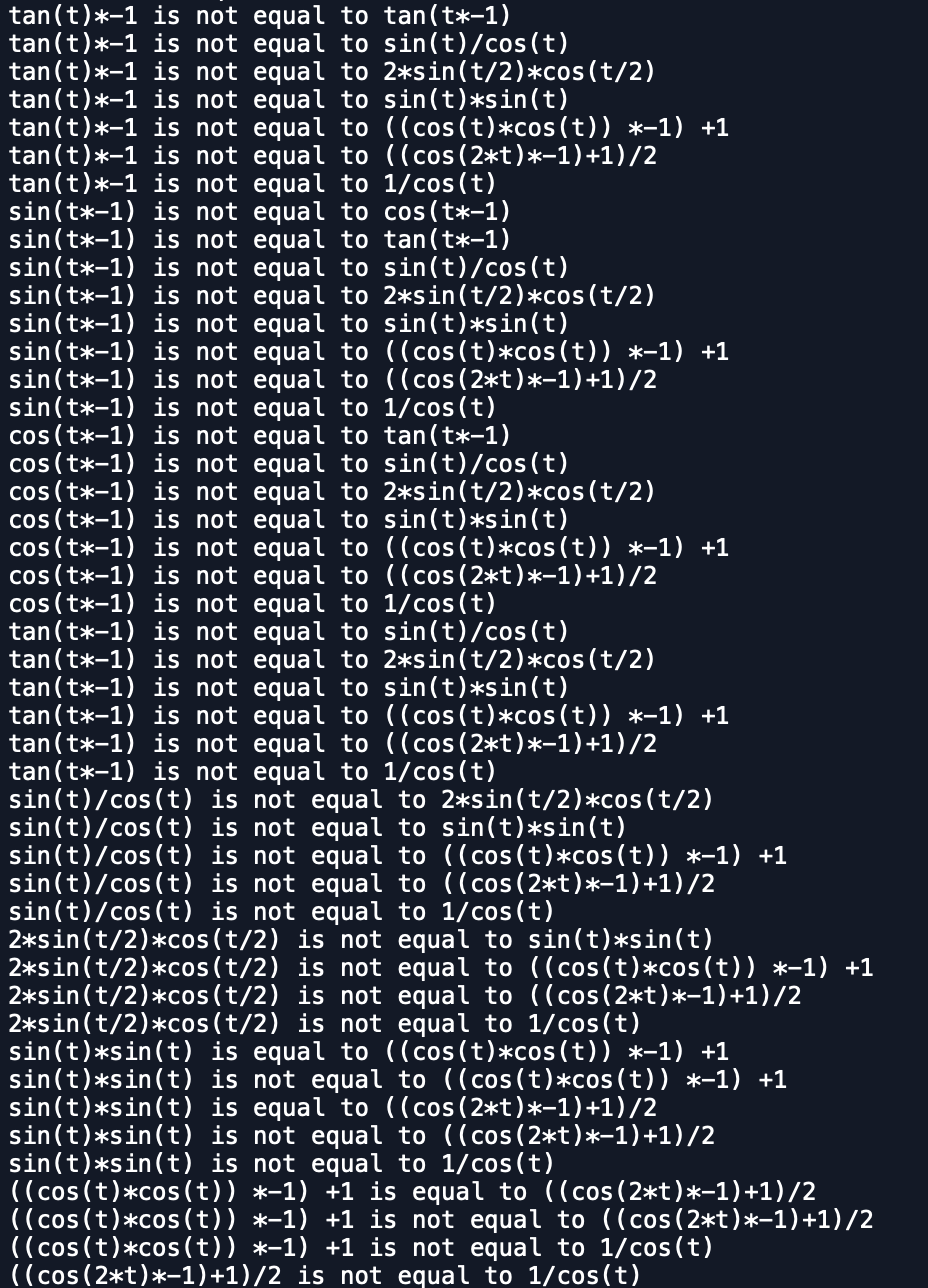
LAB 8 REPORT

In our last and final lab, we were asked to discover trig identities and test if they are equal using randomization from –pi to pi. We were also asked to partition sets into two equal subsets by finding the sum of the elements in the larger set and trying to equally divide them. The first part of the lab, the trigonometry, I found it very simple. First I have a method named isEqual which the same makes the method self-explanatory. It took the list of functions as a parameter, the method was modified by taking a range of the random numbers only from –pi to pi. Then, I have my trig method, which does most of the work. It traverses through the list of trig identities and calls my method isEqual to check if they are equal. That method being very easy to work with. Now, as far as the second part of the lab, it was a bit challenging for me. I could not get the partitioning part. I attempted by having a counter that added and totaled the sum, but after that I was not sure how to continue on to partitioning the subsets.

For my isEqual method in the trig problem, my Big O was constant, considering all it did was check if they were equal, returning True or False. My trig method seems to be O(n^2) considering there are two for loops. My results for problem 1 are as follows,







In conclusion, I learned and enjoyed working with randomization very much. I learned how to check if trig identities are equal using randomization. Although I did not finish problem 2 of this last lab, I did learn how to traverse through the sets and add to the counter finding a final sum.

#SOURCE CODE

# PAOLA TERRAZAS

# CS2302

# DISCOVER TRIG IDENTITIES USING RANDOMIZATION

# DR. FUENTES

# TA - ANINDITA NATH

# 5.10.2019

import random

import numpy as np

from mpmath import \*

import math

import mpmath

def isEqual(f1, f2 ,tries=1000, tolerance=0.0001):

for i in range(tries):

t = random.uniform(-math.pi, math.pi)

y1 = eval(f1)

y2 = eval(f2)

if np.abs(y1-y2)>tolerance:

return False

return True

def trig(trigs):

for i in range(len(trigs)):

for j in range(i+1, len(trigs)):

if isEqual(trigs[i], trigs[j]):

print(trigs[i], 'is equal to', trigs[j])

print(trigs[i], 'is not equal to', trigs[j])

trigs = ['sin(t)', 'cos(t)','tan(t)', 'sec(t)',

'sin(t)\*-1', 'cos(t)\*-1', 'tan(t)\*-1', 'sin(t\*-1)', 'cos(t\*-1)',

'tan(t\*-1)', 'sin(t)/cos(t)', '2\*sin(t/2)\*cos(t/2)', 'sin(t)\*sin(t)',

'((cos(t)\*cos(t)) \*-1) +1', '((cos(2\*t)\*-1)+1)/2', '1/cos(t)']

trig(trigs)

Scholastic Dishonesty

Any student who commits an act of scholastic dishonesty is subject to discipline. Scholastic dishonesty includes, but not limited to cheating, plagiarism, collusion, the submission for credit of any work or materials that are attributable to another person.

* **Cheating**
  + Copying form the test paper of another student
  + Communicating with another student during a test
  + Giving or seeking aid from another student during a test
  + Possession and/or use of unauthorized materials during tests (i.e. Crib notes, class notes, books, etc)
  + Substituting for another person to take a test
  + Falsifying research data, reports, academic work offered for credit
* **Plagiarism**
  + Using someone’s work in your assignments without the proper citations
  + Submitting the same paper or assignment from a different course, without direct permission of instructors
* **Collusion**
  + Unauthorized collaboration with another person in preparing academic assignments

Sign \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date: 5/10/2019