Yahir Rivas

Lab 8 Report

Instructor: Dr. Olac Fuentes

Teaching Assistant: Nath, Anindita

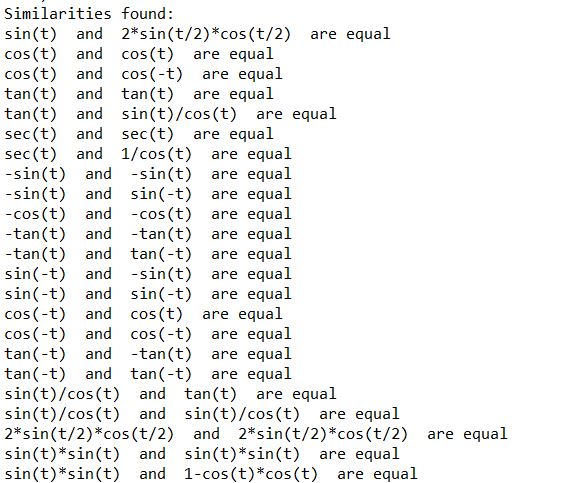
CS 2302

MW 1:30 - 2:50

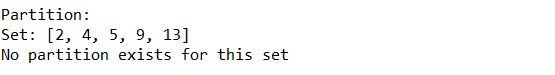
**Lab 8 Report**

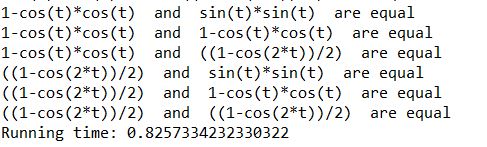
For this lab I was asked to submit a code with two different algorithms designs. One for randomization and another one for backtracking. The randomization problem consists of writing a method that compares trigonometric identities with numbers ranging from -pi to pi. While the backtracking method finds if a set has a way to be partitioned.

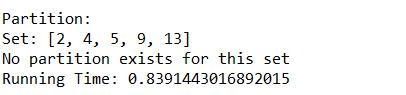
For the lab we were already given some methods by the professor which were equal to compare if two functions are equal and subset sum to compare union and disjunction. To compare the trigonometric functions is made a list with all trigonometric functions and used it as a parameter in my trigonometric comparison method. To compare trigonometric functions, I created two for loops to compare the trigonometric functions I created on the list, and using the equal method given by the professor I compared the first function to the second one.

In my partition problem I checked if a set was divisible by two if it isn’t then I return False, otherwise the sum of the set divided by two is returned. Then, to compute the subset I checked if the goal from the partition was 0 if it is I returned true and then returned an empty list if either the goal or the last index are less than 0 then I returned false and an empty list. Then I recursively called subset sum subtracting 1 from the parameter last and subtracting the goal with the last index of the set. Then that index is checked and added to one of the two subsets. Once there are two subsets the recursive calls stop or if there is no partition available.









**Conclusion**

With this Lab I learned to use backtracking and randomization while programming. Also comparing two functions is very useful and knowing how to do that was also very important for this lab.

# -\*- coding: utf-8 -\*-

"""

Created on Fri May 8 14:48:17 2019

@author: Yahir F. Rivas

"""

import random

import numpy as np

from math import \*

import time

def equal(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 compare\_trig(f):

for i in range(len(f)):

for j in range(1,len(f)):

if equal(f[i], f[j],tries=1000,tolerance=0.0001):

print(f[i], " and ", f[j], " are equal")

def subsetsum(S,last,goal):

if goal == 0:

return True, []

if goal<0 or last<0:

return False, []

res, subset = subsetsum(S,last-1,goal-S[last])

if res:

subset.append(S[last])

return True, subset

else:

return subsetsum(S,last-1,goal)

def partition(S):

if sum(S)%2 != 0:

return False

goal = sum(S)//2

return goal

functions = ['sin(t)', 'cos(t)', 'tan(t)', 'sec(t)', '-sin(t)', '-cos(t)',

'-tan(t)', 'sin(-t)','cos(-t)', 'tan(-t)', 'sin(t)/cos(t)',

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

print('Similarities found:')

compare\_trig(functions)

print()

S = [2,4,5,9,13]

print('Partition:')

print('Set:' , S)

part = partition(S)

boo, sub1= subsetsum(S,len(S)-1,part)

if part:

sub2 = []

print('First subset:',sub1)

for j in S:

if j not in sub1:

sub2.append(j)

print('Second subset:',sub2)

else:

print('No partition exists for this set')

I certify that this project is entirely my own work. I wrote, debugged, and tested the code being presented, performed the experiments, and wrote the report. I also certify that I did not share my code or report or provided inappropriate assistance to any student in the class.