Erick Perchez

80582912

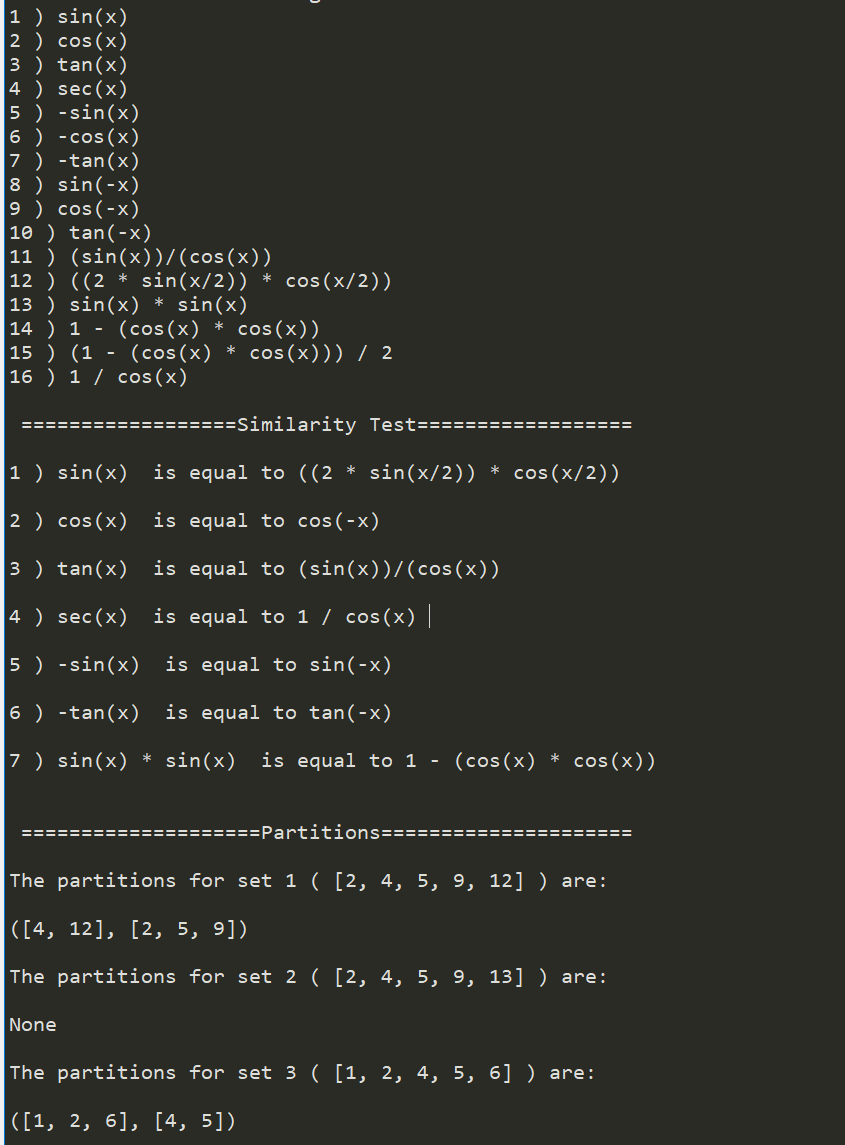
CS 2302 Data Structures

5/14/2019

Lab 8

**Introduction:** In this lab I find similarities between two trig functions and to find equal subsets of the same set.

**Proposed solution:**

1. Write a program to discover trig identities: To do this, I took all the given functions and implemented them in a list as strings (as python can recognize them). I then pass them through a method called equalTrig. This method starts with one for loop that loops for the length of the list of equations. The second forloop starts at I + 1 and ends the length of the list of equations. Given the method equal, I checked if equation I was the same as equation j. if they were, I display the function and say that it is equivalent to function j. Running Time: O(n^2)
2. Partition problem: To do this, I created a method called equalPartition. This partition takes a list of numbers as a parameter. It starts with a conditional that states that if the sum of all the numbers is even, if so, the conditional goes through. I initialize a variable to be the sum of the set. I then call a method, subsetsum, given to us and pass the Set, length -1, and the variable total divided by two. I set the returned values to x and set1 I then create a second empty list to be the second set. I then create a for loop with a conditional inside that states if S[i] is not in set1 then I append S[i] to set 2. After the for loop theres another conditional, if the sum of set 1 and set 2 are not the same, then I return none. I finally return the two sets.
3. **Experimental Results:**

**Conclusion:** I learned a lot of different ways to involve different algorithms in my code. It was tough but it seemed to work

**Appendix:**

'''

Course: CS2302

Author: Erick Perchez

Assignment: Lab 8

Instructor: Dr. Fuentes

TA: Andita Nath

Date: 05/12/2019

Purpose: To find similarities between two trig functions and to find

equal subsets of the same set.

'''

import random

import numpy as np

from mpmath import \*

from math import \*

def equal(f1, f2,tries=1000,tolerance=0.01):

for i in range(tries):

x = random.random()

y1 = eval(f1)

y2 = eval(f2)

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

return False

return True

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]) # Take S[last]

if res:

subset.append(S[last])

return True, subset

else:

return subsetsum(S,last-1,goal) # Don't take S[last]

''' #1

================================================================

'''

#Takes list of equations and compares all items on the list with

#each other. The list sees the list of trigonometric functions

#then i check their values and compare them using the given equal

#method to see if theyre similar.

def equalTrig(eq):

number = 1

for i in range(len(eq)):

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

if equal(eq[i], eq[j]) == True:

print(number, ')', eq[i], ' is equal to', eq[j],'\n')

number += 1

''' #2

================================================================

'''

#Takes a set of numbers and splits into two different sets containing

#the same numbers but their sums are equal. Given the subsetsum

#method given i will be able to get two equal value sets.

#I then check if their sums are similar, if so i return both.

def equalPartition(S):

if sum(S) % 2 == 0:

total = sum(S)

x, set1 = subsetsum(S, len(S) - 1, total/2)

set2 = []

for i in range(len(S)):

if S[i] not in set1:

set2.append(S[i])

if sum(set1) != sum(set2):

return None

return set1, set2

return None

''' M A I N

================================================================

'''

tList = ['sin(x)', 'cos(x)', 'tan(x)', 'sec(x)','-sin(x)', '-cos(x)', '-tan(x)', 'sin(-x)', 'cos(-x)',

'tan(-x)', '(sin(x))/(cos(x))', '((2 \* sin(x/2)) \* cos(x/2))', 'sin(x) \* sin(x)', '1 - (cos(x) \* cos(x))',

'(1 - (cos(x) \* cos(x))) / 2', '1 / cos(x)']

print('\n =====================Trig List=====================')

for i in range(len(tList)):

print(i+1,')',tList[i])

print('\n ==================Similarity Test==================\n')

equalTrig(tList)

print('\n ====================Partitions=====================\n')

set1 = [2, 4, 5, 9, 12]

set2 = [2, 4, 5, 9, 13]

set3 = [1, 2, 4, 5, 6]

print('The partitions for set 1 (', set1, ')', 'are:\n')

print(equalPartition(set1), '\n')

print('The partitions for set 2 (', set2, ')', 'are:\n')

print(equalPartition(set2), '\n')

print('The partitions for set 3 (', set3, ')', 'are:\n')

print(equalPartition(set3), '\n')

“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”