Samuel Chong

80623443

CS 2302

Lab 8

Olac Fuentes

Introduction: ﻿Purpose was to discover if the trigonometric identities are equal to others

and to use backtracking in a set and determine if there is partition.

Methods: For this lab I used 2 methods that the professor provided us. Equal and SubsetSum were the methods given by professor, the equal checks if the trigonometric functions that are parameters are equal or not, the method has a 0.0001 tolerance of value to check if they are equal values. subsetSum return True if the goal is equal to 0, returns false when goal or last is less than 0. If set is able to do partition then the method returns the two subsets.

Partition checks if the sum of the set % 2 is not equal to 0 then return “No partition exists”, if it is equal to 0 then create a new list with the copy of the set given as a parameter and then check if they are equal print the two sub sets, else return.

Output:



|  |  |
| --- | --- |
| Methods | Running Time |
| Equal | ﻿2.3362989 sec |
| Partition | ﻿7.4148178 sec |

Code:

﻿#!/usr/bin/env python3

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

"""

@author: Samuel Chong

MW 10:30-11:50

Olac Fuentes

TAs: Anindita Nath and Maliheh Zargaran

Purpose was to discover if the trigonometric identities are equal to others

and to use backtracking in a set and determine if there is partition

"""

import random

import numpy as np

from mpmath import \*

import time

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

for i in range(tries):

t = random.random()

y1 = eval(f1)

y2 = eval(f2)

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

return False

return True

def subsetsum(S,S1,last,goal):

if goal ==0:

return True, []

if goal<0 or last<0:

return False, []

res, subset = subsetsum(S,S1,last-1,goal-S[last]) # Take S[last]

if res:

subset.append(S[last]) #append to one subset

S1.remove(S[last]) #remove from S1 because it is part from the other subset

return True, subset

else:

return subsetsum(S,S1,last-1,goal) # Don't take S[last]

def equalTrig(trig):

for i in range(len(trig)):

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

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

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

def partition(S):

if sum(S) %2 != 0:

print("No partition exists")

if sum(S) %2 == 0:

newSet = S.copy() #create a copy of S

e, rem = subsetsum(S,newSet,len(S)-1,sum(S)//2) #check if equal

if e:

print("Subset 1: ",newSet)

print("Subset 2: ",rem)

return

trigFunc = ['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)', '1/cos(t)']

start= time.time()

for i in range(len(trigFunc)):

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

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

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

end = time.time()

print()

print("Time it takes to know if the funtions are equal: ", end - start, "seconds")

print()

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

print("Set is: ", S)

start = time.time()

partition(S)

end = time.time()

print("Time it takes use partition: ", end - start, "seconds")

I, Samuel Chong, sign the academic honesty certification. This is my work and only my work. No external help was used for this lab. Also, this report was made by me and no collaboration was made.