CS 2302

Lab 8 Report

Author: David Ayala

ID: 80617838

Instructor: Olac Fuentes

TA: Anindita Nath

**Introduction**

The purpose of this lab was to (Randomized algorithms) Write a program to ”discover” trigonometric identities. Your program should test all combinations of the trigonometric expressions shown below and use a randomized algorithm to detect the equalities. (Backtracking) The partition problem consists of determining if there is a way to partition a set of integers. This is to see if you can do Randomized, Backtracking algorithms.

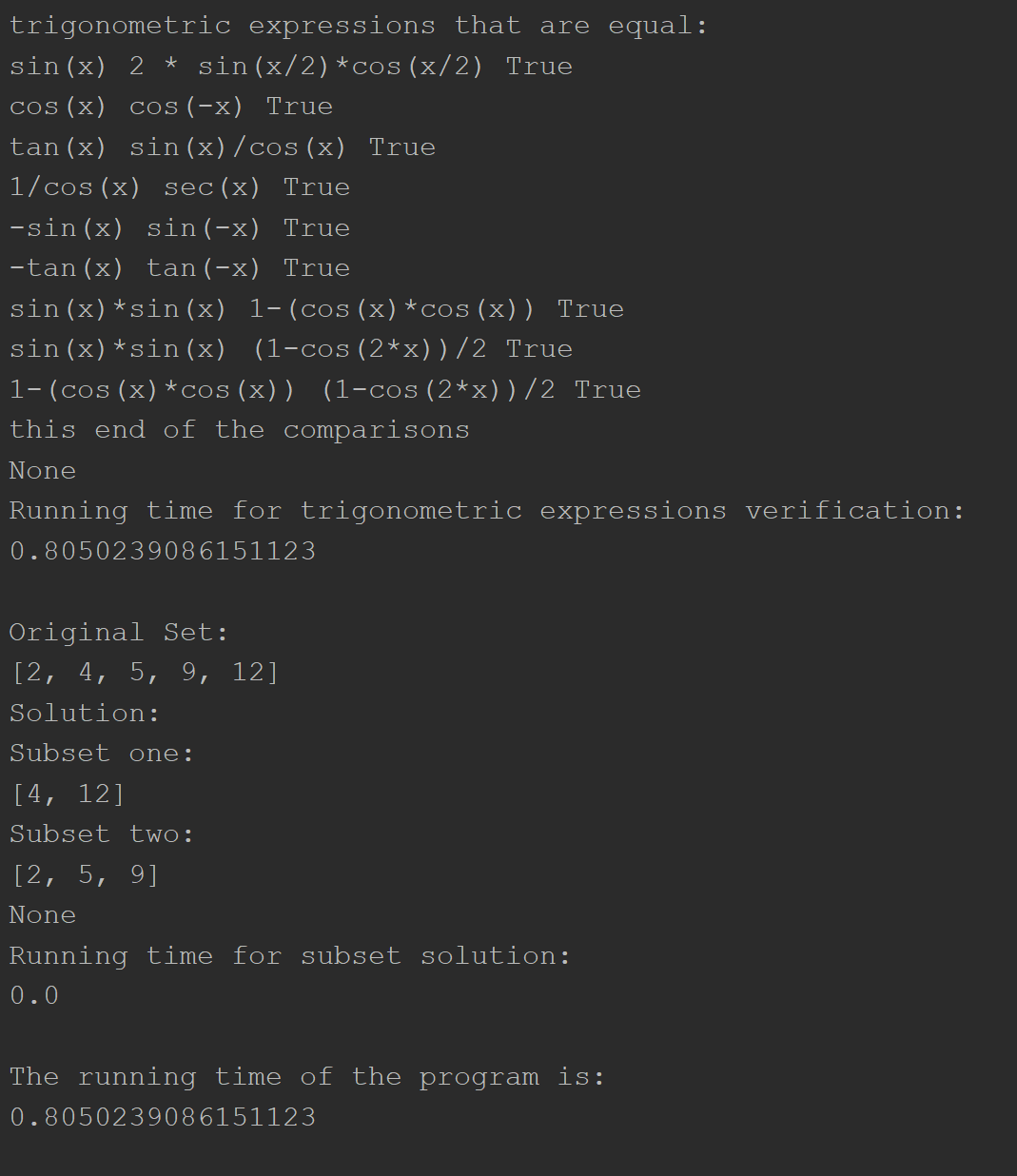
**Proposed Solution & Design Implementation**

What I did to do this lab was first used the code that was on professor Fuentes website “SubSum”,

“Equal”. I didn’t change “equal”, but I did modify “Subsum” to take another parameter S2 that way there could be two subsets. Once that was done, I then created a function to be able to traverse the list of the Trig expressions and be able to compare them use the “Equal”. I then created a function to be able to see if the set has a solution and to be able to partition if there is a solution. Honestly a lot of this lab was just either reusing his code or modifying it to match what you need it to do.

**Experimental Results**

|  |  |
| --- | --- |
| **section** | **Running Times** |
| **trigonometric expressions verification** | 0.8050239086151123 |
| **subset solution** | 0.0 (too small to make a significant difference) |
| **Entire program** | 0.8050239086151123 |

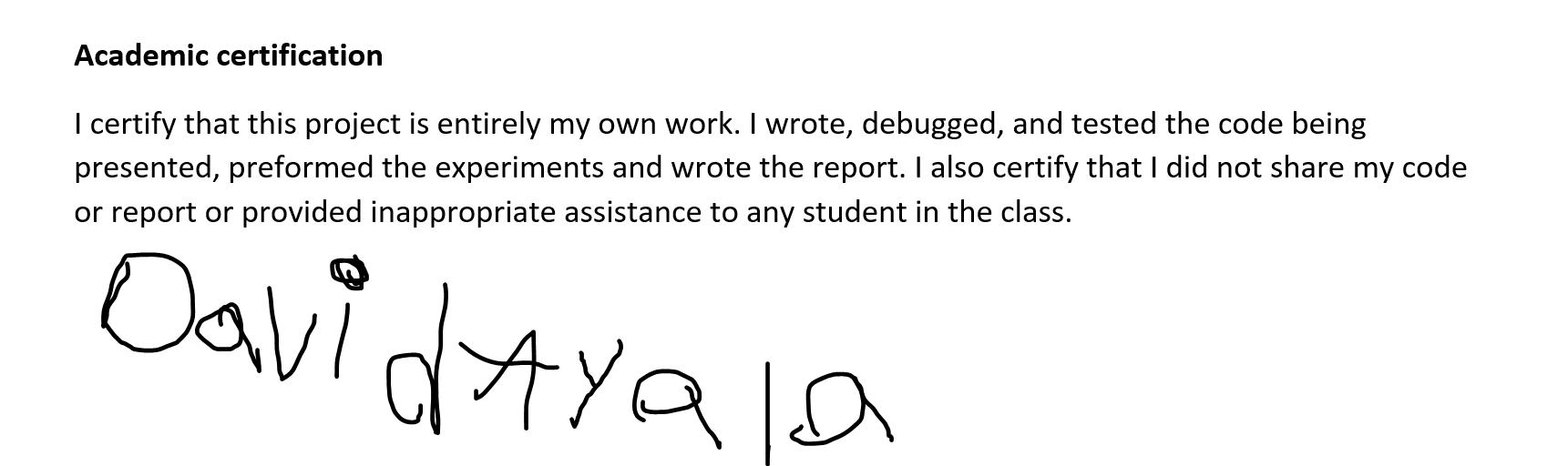


**Conclusion**

This lab allowed us to learn how to Randomized, Backtracking algorithms. See how they can be used in the real world. This lab also was great way to brush up Randomized, Backtracking algorithms, as well as other various code that we may have done before.

**Appendix**

# Course:CS 2302 MW 1:30-2:50, Author:David Ayala  
# Assignment:Lab #8, Instructor: Olac Fuentes  
# Teaching Assistant: Maliheh Zargaran, Date of last Modification: 5/12/2019  
# Purpose of program:. (Randomized algorithms) Write a program to  
# ”discover” trigonometric identities. Your program should  
# test all combinations of the trigonometric expressions shown below  
# and use a randomized algorithm to detect the equalities.  
# (Backtracking) The partition problem consists of determining  
# if there is a way to partition a set of integers.  
  
  
from mpmath import \*  
import numpy as np  
import random  
import time  
  
# code from fuentes' website to see if  
# two given arguments are the same  
def equal(F1, F2, tries = 1000, tolerance=0.0001):  
 for i in range(tries):  
 x = random.random()  
 y1 = eval(F1)  
 y2 = eval(F2)  
 if np.abs(y1-y2) > tolerance:  
 return False  
 return True  
  
# allows the traversal of the list that why it can  
# compare the two trigonometric identities  
def equationVerification(TrigList):  
 for i in range(len(TrigList)):  
 for j in range(i + 1, len(TrigList), 1):  
 if equal(TrigList[i],TrigList[j]): print(TrigList[i], TrigList[j], True)  
 print('this end of the comparisons')  
  
TrigList = ['sin(x)', 'cos(x)', 'tan(x)', '1/cos(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(2\*x))/2', 'sec(x)']  
  
# code from fuentes' website expect modified to take a S2  
# to allow for backtracking  
def SubSum(S,S2, length, goal):  
 if goal == 0:  
 return True, []  
 if goal < 0 or length < 0:  
 return False, []  
 res, subset = SubSum(S, S2, length - 1, goal-S[length])  
 if res:  
 subset.append(S[length])  
 S2.remove(S[length])  
 return True, subset  
 else:  
 return SubSum(S, S2, length-1, goal)  
# code allows the partition of the set and then  
# checks to see if there is a solution  
def PartitionOfTheSet(S):  
 summation = sum(S)  
 if summation % 2 != 0:  
 print('No Solution')  
 return False  
 elif summation % 2 == 0:  
 S2 = [i for i in S]  
 temp, s = SubSum(S, S2, len(S)-1, sum(S)//2)  
 if temp:  
 print('Original Set:')  
 print(S)  
 print('Solution:')  
 print('Subset one:')  
 print(s)  
 print('Subset two:')  
 print(S2)  
 else:  
 print('No Solution')  
 return False  
  
# The following two lines of code are the  
# two different test case that fuentes had  
# on the lab description the fist S is the  
# one that will give solution while the second  
# S is the one that will not give you a Solution  
S = [2, 4, 5, 9, 12]  
# S = [2, 4, 5, 9, 13]  
S2 = []  
  
Start = time.time()  
  
Start2 = time.time()  
print('trigonometric expressions that are equal:')  
print(equationVerification(TrigList))  
end2 = time.time()  
total2 = end2 - Start2  
print('Running time for trigonometric expressions verification:')  
print(total2)  
print()  
  
Start3 = time.time()  
print(PartitionOfTheSet(S))  
end3 = time.time()  
total3 = end3 - Start3  
print('Running time for subset solution:')  
print(total3)  
  
end = time.time()  
total = end - Start  
print()  
print('The running time of the program is:')  
print(total)



David Ayala

5/12/19