Introduction:

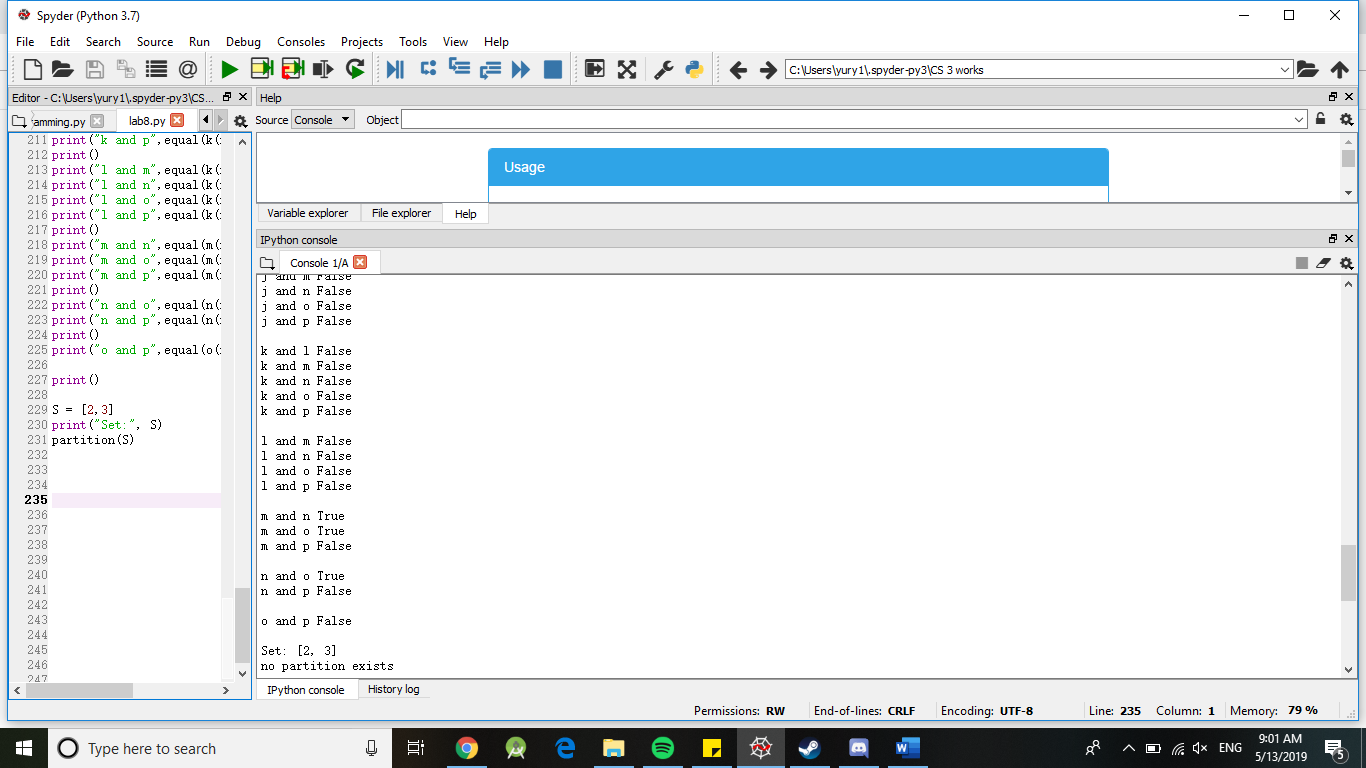
The problem that was to be solved was to make a maze out of disjoint set forests and apply different searches to it

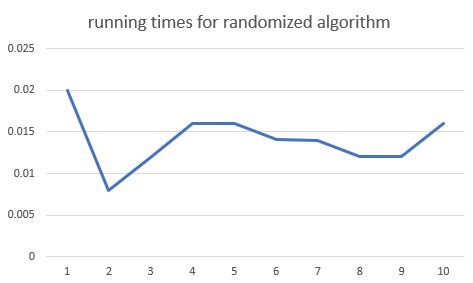
Solution:

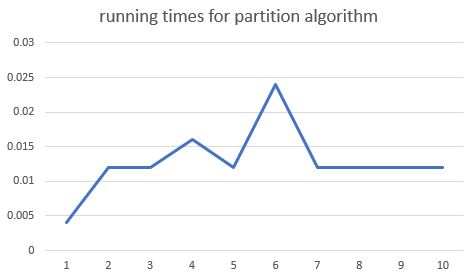
I first attempted at the problem by writing pseudocode and trying to make sense of it. I then implemented it into code and ran several times. Each time I ran it I got close to solving the problem. I finally followed it up with actually coming up with the solution.

Results:

The experiments included running the program several times, and finding out the outputs. The running times.







Conclusions:

I learned that disjoint set forests are very useful.

I Yury Ionov 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.



Source code:

import random

import math

import mpmath

import numpy as np

def equal(f1, f2):

y1 = eval('f1')

y2 = eval('f2')

if np.abs(y1-y2)>.00000001:

return False

return True

def a(x):

return math.sin(x)

def b(x):

return math.cos(x)

def c(x):

return math.tan(x)

def d(x):

return mpmath.sec(x)

def e(x):

return -(math.sin(x))

def f(x):

return -(math.cos(x))

def g(x):

return -(math.tan(x))

def h(x):

return math.sin(-x)

def i(x):

return math.cos(-x)

def j(x):

return math.tan(-x)

def k(x):

return (math.sin(x)/math.cos(x))

def l(x):

return 2\*(math.sin(x//2)\*math.cos(x//2))

def m(x):

return math.pow(math.sin(x),2)

def n(x):

return 1 - math.pow(math.cos(x),2)

def o(x):

return (1-math.cos(2\*x))/2

def p(x):

return 1/math.cos(x)

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]

def partition(S):

tot=0

for i in S:

tot = tot + i

true = False

if tot%2==0:

goal = tot//2

true,partSet = subsetsum(S,len(S)-1,goal)

if true == True:

print(partSet)

true,partSet2 = subsetsum(S,len(S)-2,goal)

print(partSet2)

else:

print("no partition exists")

##############################################################33

x = random.random()

x = (x \* (2\*math.pi))-math.pi

print("a and b",equal(a(x),b(x)))

print("a and c",equal(a(x),c(x)))

print("a and d",equal(a(x),d(x)))

print("a and e",equal(a(x),e(x)))

print("a and f",equal(a(x),f(x)))

print("a and g",equal(a(x),g(x)))

print("a and h",equal(a(x),h(x)))

print("a and i",equal(a(x),i(x)))

print("a and j",equal(a(x),j(x)))

print("a and k",equal(a(x),k(x)))

print("a and l",equal(a(x),l(x)))

print("a and m",equal(a(x),m(x)))

print("a and n",equal(a(x),n(x)))

print("a and o",equal(a(x),o(x)))

print("a and p",equal(a(x),p(x)))

print()

print("b and c",equal(b(x),c(x)))

print("b and d",equal(b(x),d(x)))

print("b and e",equal(b(x),e(x)))

print("b and f",equal(b(x),f(x)))

print("b and g",equal(b(x),g(x)))

print("b and h",equal(b(x),h(x)))

print("b and i",equal(b(x),i(x)))

print("b and j",equal(b(x),j(x)))

print("b and k",equal(b(x),k(x)))

print("b and l",equal(b(x),l(x)))

print("b and m",equal(b(x),m(x)))

print("b and n",equal(b(x),n(x)))

print("b and o",equal(b(x),o(x)))

print("b and p",equal(b(x),p(x)))

print()

print("c and d",equal(c(x),d(x)))

print("c and e",equal(c(x),e(x)))

print("c and f",equal(c(x),f(x)))

print("c and g",equal(c(x),g(x)))

print("c and h",equal(c(x),h(x)))

print("c and i",equal(c(x),i(x)))

print("c and j",equal(c(x),j(x)))

print("c and k",equal(c(x),k(x)))

print("c and l",equal(c(x),l(x)))

print("c and m",equal(c(x),m(x)))

print("c and n",equal(c(x),n(x)))

print("c and o",equal(c(x),o(x)))

print("c and p",equal(c(x),p(x)))

print()

print("d and e",equal(d(x),e(x)))

print("d and f",equal(d(x),f(x)))

print("d and g",equal(d(x),g(x)))

print("d and h",equal(d(x),h(x)))

print("d and i",equal(d(x),i(x)))

print("d and j",equal(d(x),j(x)))

print("d and k",equal(d(x),k(x)))

print("d and l",equal(d(x),l(x)))

print("d and n",equal(d(x),n(x)))

print("d and m",equal(d(x),m(x)))

print("d and o",equal(d(x),o(x)))

print("d and p",equal(d(x),p(x)))

print()

print("f and g",equal(f(x),g(x)))

print("f and h",equal(f(x),h(x)))

print("f and i",equal(f(x),i(x)))

print("f and j",equal(f(x),j(x)))

print("f and k",equal(f(x),k(x)))

print("f and l",equal(f(x),l(x)))

print("f and m",equal(f(x),m(x)))

print("f and n",equal(f(x),n(x)))

print("f and o",equal(f(x),o(x)))

print("f and p",equal(f(x),p(x)))

print()

print("g and h",equal(g(x),h(x)))

print("g and i",equal(g(x),i(x)))

print("g and j",equal(g(x),j(x)))

print("g and k",equal(g(x),k(x)))

print("g and l",equal(g(x),l(x)))

print("g and m",equal(g(x),m(x)))

print("g and n",equal(g(x),n(x)))

print("g and o",equal(g(x),o(x)))

print("g and p",equal(g(x),p(x)))

print()

print("h and i",equal(h(x),i(x)))

print("h and j",equal(h(x),j(x)))

print("h and k",equal(h(x),k(x)))

print("h and l",equal(h(x),l(x)))

print("h and m",equal(h(x),m(x)))

print("h and n",equal(h(x),n(x)))

print("h and o",equal(h(x),o(x)))

print("h and p",equal(h(x),p(x)))

print()

print("i and j",equal(i(x),j(x)))

print("i and k",equal(i(x),k(x)))

print("i and l",equal(i(x),l(x)))

print("i and m",equal(i(x),m(x)))

print("i and n",equal(i(x),n(x)))

print("i and o",equal(i(x),o(x)))

print("i and p",equal(i(x),p(x)))

print()

print("j and k",equal(j(x),k(x)))

print("j and l",equal(j(x),l(x)))

print("j and m",equal(j(x),m(x)))

print("j and n",equal(j(x),n(x)))

print("j and o",equal(j(x),o(x)))

print("j and p",equal(j(x),p(x)))

print()

print("k and l",equal(k(x),l(x)))

print("k and m",equal(k(x),m(x)))

print("k and n",equal(k(x),n(x)))

print("k and o",equal(k(x),o(x)))

print("k and p",equal(k(x),p(x)))

print()

print("l and m",equal(k(x),p(x)))

print("l and n",equal(k(x),n(x)))

print("l and o",equal(k(x),o(x)))

print("l and p",equal(k(x),p(x)))

print()

print("m and n",equal(m(x),n(x)))

print("m and o",equal(m(x),o(x)))

print("m and p",equal(m(x),p(x)))

print()

print("n and o",equal(n(x),o(x)))

print("n and p",equal(n(x),p(x)))

print()

print("o and p",equal(o(x),p(x)))

print()

S = [2,3]

print("Set:", S)

partition(S)