Michelle Song
Chen Ren
Hw 4

Problem Statement

The Graph Problem

1. Design an algorithm that finds paths through a given graph. Provide step-by-step data for an example problem. You will be using that step-by-step solution for debugging and validation.
2. Implement and document a program that finds paths through a given graph. Use the posted slideshows and follow the given problem definition. Include the algorithm from part 1 in your documentation.
3. Analyze possible paths on paper and demonstrate that your code finds all existing, no duplicates, and no false ones.
4. Upload the documentation (including proof of successful implementation) as a single document/report.

Algorithm

Main.py

1. Find folder path for graph folder (which contains lines.txt and nodes.txt)
2. Call Graph.py with folder path (g=Graph(folder path))
3. Call strings from nodes.py and lines.py
4. Find paths
    a. For i in g.nodes.keys() (nodes is a dictionary)
        i. For j in g.nodes.keys()
            1. PathList = g.findPath(i, j) (Graph.py)
            2. Print each path
        ii. Find shortest path (Graph.py)

Graph.py

1. Define class Graph(object)
2. Defin __init__ (self, folder_name)
    a. Define a dictionary to hold Nodes
    b. Read in Nodes text file and put nodes into Nodes (dictionary)

Michelle Song
Chen Ren
Hw 4

   c. Define a lines dictionary
   d. Read in lines.txt
      i. Use ID of the 2 nodes that each line connects and put into lines dictionary
      ii. Calculate distance between the two nodes of the line (define as length)
      iii. Lines dictionary = {LineID, node1ID, node2ID, length)
3. Define __str__
   a. Define list of nodes
   b. Define list of lines
   c. Append nodes to nodes list
   d. Append lines to lines list
4. Define Find Path
   a. If StartID not in nodes.keys
      i. Print "unknown nodeID"
   b. If EndID not in nodes.key
      i. Print "unknown nodeID"
   c. Define a global variable path list (as a list)
   d. Define path as a dictionary (nodepath, linepath, length)
   e. Call findpathinside (next section)
5. Define findpathinside
   a. Use startID to initiate path finding
   b. Find lines that originate from start ID and the nodes the lines go to
   c. Repeat
      i. For each node in path, find all lines that originate from node
      ii. Record the node that each lines goes to
      iii. If node = previous node, then do not record and continue to next line/node
      iv. Record length of each line traveled
      Length += previous Length
      v. Repeat until no more possible lines/nodes
6. Define shortest path
   a. Find minimum path length using
      i. For i in range(all paths)
         1. If new path length < current path length
            a. Record path as shortest
         2. Else
            a. Continue onto next path
         3. Repeat until all paths are tested
7. Define numnodes
   a. Returns number of nodes
8. Define numlines
   a. Return number of lines

Michelle Song
Chen Ren
Hw 4

Node.py

1. Define __init__
    a. Define id
    b. Define lines list
    c. Call vector.py as node
2. Define get position as node
3. Define Node ID
4. Define Attach
    a. Attach lineID
5. Define detach
    a. Detach lineID
6. Define str
    a. Return ID,Nodes, NodeinlinesID
7. Define attachedlines as lines

Line.py

1. Define init
    a. Define nodeIDs, LineIDs, list of nodes, length of line
2. Define len as length of line
3. Define getID as lineID
4. Define getNodes as Node
5. Define Attach
    a. If nodeid not in self.nodes
        i. Append nodeid
6. Define Detach
    a. If nodeID in self.nodes and len(self.nodes)>0
    b. Remove nodeID
7. Def length as self.length

Vector.py

1. Define init as self.v=data
2. Define getitem
3. Define setitem
4. Define len
5. Define vector addition
6. Define vector subtraction
7. Define vector multiplication
8. Define vector division as not implemented
9. Define distance (vector travels)

Michelle Song
Chen Ren
Hw 4

Implementation

Code

Main

```
from Graph import *

folder_graph1= '/Users/helloRC/eclipse-workspace/cee505 hw1/src/hw4/Example Input22/Graph1'
folder_graph2= '/Users/helloRC/eclipse-workspace/cee505 hw1/src/hw4/Example Input22/Graph2'
g = Graph(folder_graph1)
g2 = Graph(folder_graph2)

print g,"\n"
for i in g.nodes.keys():
   for j in g.nodes.keys():
      if i!=j:
         pathList = g.findPath(i,j)
         for path in pathList:
            print "The length from nodeID {} to nodeID {} is {}, passing the lineID{}".format(i,j,path['length'],path['lines_path'])
         shortpath = g.findShortestPath(i,j)
         print "The shortest path above is {},which passes the nodesID {} and the linesID {}".format(shortpath['length'],shortpath['nodes_path'],shortpath['lines_path']),'\n'

print g2,"\n"
for i in g2.nodes.keys():
   for j in g2.nodes.keys():
      if i!=j:
         pathList = g2.findPath(i,j)
         for path in pathList:
            print "The length from nodeID {} to nodeID {} is {}, passing the lineID{}".format(i,j,path['length'],path['lines_path'])
         shortpath = g2.findShortestPath(i,j)
         print "The shortest path above is {},which passes the nodesID {} and the linesID {}".format(shortpath['length'],shortpath['nodes_path'],shortpath['lines_path']),'\n'
```

Graph

```
import os
from Graph import *
from Line import *
from Node import *
from docutils.parsers.rst.directives import path
```

Michelle Song
Chen Ren
Hw 4

```
from copy import deepcopy

class Graph(object):
    def __init__(self, folder_name):
        os.chdir(folder_name )
        self.nodes = {}
        f = open('nodes.txt', 'r')
        for line in f:
            l = line.rstrip().split('\t')
            v = Vector([float(l[1]), float(l[2])])
            self.nodes[l[0]]= Node(l[0],v)
        f.close()

        self.lines = {}
        f = open('lines.txt', 'r')
        for line in f:
            l = line.rstrip().split('\t')
            for nodeid in self.nodes.keys():
                if nodeid==l[1]:
                    self.nodes[nodeid].attach(l[0])
                    p1 = self.nodes[nodeid].getPosition()
                elif nodeid==l[2]:
                    self.nodes[nodeid].attach(l[0]);
                    p2 = self.nodes[nodeid].getPosition()
                else:
                    "The nodeID doesn't match"
            length=(Vector(p1-p2)).distance()
            length = round(length, 2)
            self.lines[l[0]] = Line(l[0], l[1], l[2],length)

        f.close()

    def __str__(self):
        Nodes_list = []
        Lines_list = []
        for i in self.nodes:
            Nodes_list.append(self.nodes[i].__str__())
        for i in self.lines:
            Lines_list.append(self.lines[i].__str__())
        return 'Nodes ='+'\n'.join(Nodes_list) +'\n'+'\n'+'Lines = '+'\n'.join(Lines_list)

    def findPath(self, startID, endID):
        global PathList
        PathList = []
```

Michelle Song
Chen Ren
Hw 4

```
        if startID not in self.nodes.keys() :
            print "unknown nodeid={}".format(startID)
        if endID not in self.nodes.keys():
            print "unknown nodeid={}".format(endID)

        path = dict(nodes_path = [], lines_path = [], length = 0.0)
        self.DFS(str(startID), str(endID), path, Nodes_Traveled = [], Lines_Traveled = [])
        return PathList

    def DFS(self, startID, endID, path,Nodes_Traveled, Lines_Traveled):
        if startID not in Nodes_Traveled:
            traveledNodes2 = deepcopy(Nodes_Traveled)
            traveledLines2 = deepcopy(Lines_Traveled)
            traveledNodes2.append(startID)
            for l in self.nodes[startID].getAttachedLines():
                if l not in traveledLines2:
                    traveledLines2.append(l)
                    for n in self.lines[l].getNodes():
                        if n not in traveledNodes2:
                            pathreal = deepcopy(path)
                            pathreal['nodes_path'].append(startID)
                            pathreal['lines_path'].append(l)
                            pathreal['length']+=self.lines[l].length
                            if n==endID:
                                pathreal['nodes_path'].append(endID)
                                PathList.append(pathreal)
                            else:
                                self.DFS(n, endID, pathreal,traveledNodes2, traveledLines2)

    def findShortestPath(self, startID, endID):
        Paths = self.findPath(startID, endID)
        shortest_path = Paths[0]['length']
        j=0
        for i in range(1,len(Paths)):
            if Paths[i]['length'] < shortest_path:
                j=i
                shortest_path = Paths[i]['length']
        return Paths[j]

    def numNodes(self):
        return len(self.nodes)

    def numLines(self):
        return len(self.lines)
```

Michelle Song
Chen Ren
Hw 4

Node

```
from Vector import *

class Node(object):

    def __init__(self, id, v):
        self.id = id
        self.node = Vector(v)
        self.lines = []

    def getPosition(self):
        return self.node

    def getID(self):
        return self.id

    def attach(self, lineID):
        self.lines.append(lineID)

    def detach(self, lineID):
        self.lines.remove(lineID)

    def __str__(self):
        return "\nID:{}, Nodes:{}, NodeInLinesID:{}".format(self.id,self.node,self.lines)

    def getAttachedLines(self):
        return self.lines
```

Line

```
class Line(object):

    def __init__(self, id, nodeIid, nodeJid, length):
        self.nodeIid = nodeIid
        self.nodeJid = nodeJid
        self.id = id
        self.nodes = [nodeIid, nodeJid]
        self.nodes.sort()
        self.length = length

    def __len__(self):
        return self.length

    def getID(self):
        return self.id
```

Michelle Song
Chen Ren
Hw 4

```
  def getNodes(self):
    return self.nodes

  def attach(self, nodeID):
    if nodeID not in self.nodes:
      self.nodes.append(nodeID)
      return self.nodes
    else:
      return "{} is already in the Node.".format (nodeID)

  def detach(self, nodeID):
    if nodeID in self.nodes and len(self.nodes)>0:
      self.nodes.remove(nodeID)
      return self.nodes
    if nodeID not in self.nodes:
      return "{} is not in the Node".format(nodeID)

  def __str__(self):
    return "\nID:{}, NodeStart:{}, NodeEnd:{}".format(self.id,self.nodeId,self.nodeJid)

  def length(self):
    return self.length
```

Vector

```
import math
class Vector(object):

  def __init__(self, data):
    self.v = data

  def __getitem__(self, i):
    return self.v[i]

  def __setitem__(self, i, val):
    self.v[i] = val

  def __len__(self):
    return len(self.v)

  def __add__(self, x):
    if type(x) == Vector:
      w = self.v[:]
      for i in range(0,len(self)):
```

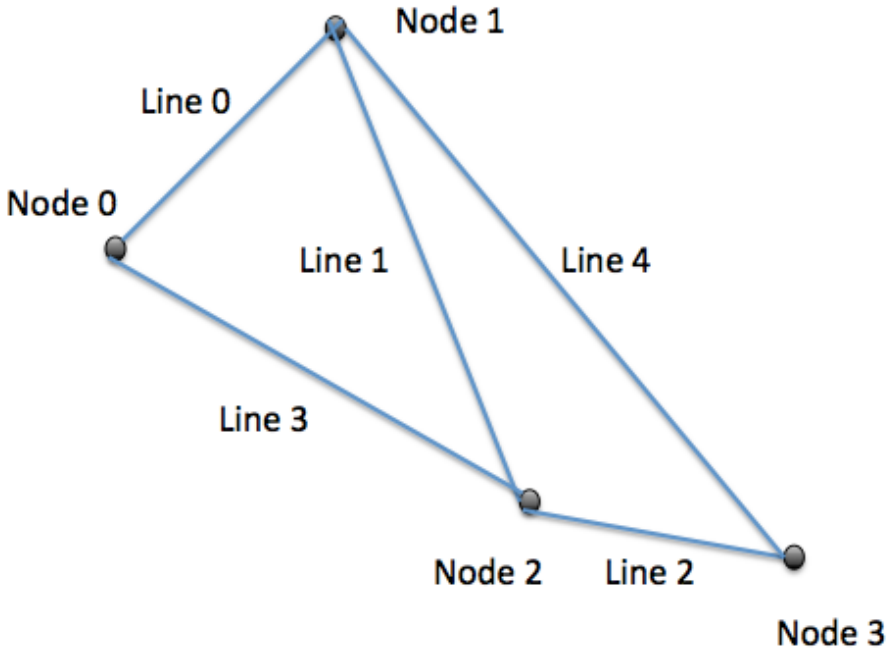
Michelle Song
Chen Ren
Hw 4

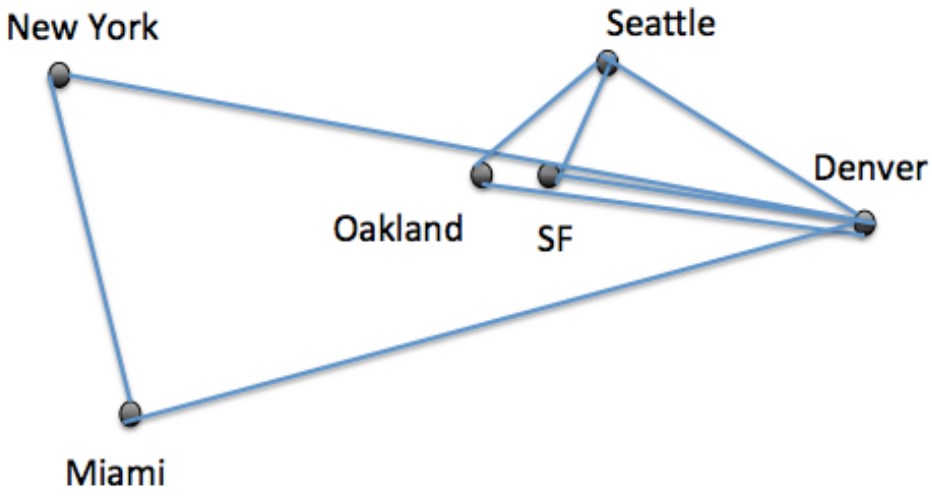
```
            w[i] += x[i]
        return w
      else:
        raise TypeError

  def __sub__(self, x):
    if type(x) == Vector:
      w = self.v[:]
      for i in range(0,len(self)):
        w[i] -= x[i]
      return w
    else:
      raise TypeError

  def __mul__(self, x):
    if type(x) == Vector:
      w = 0.0
      for i in range(0,len(self)):
        w += self.v[i] * x[i]
      return w
    else:
      raise TypeError

  def __div__(self, x):
    raise NotImplemented

  def __str__(self):
    return str( self.v )

  def __repr__(self):
    return self.__str__()

  def distance(self):
    return math.sqrt((self.v[0]*self.v[0] + self.v[1]*self.v[1]))
```

Output

We compared our python output to our drawn graphs.

Michelle Song
Chen Ren
Hw 4





Graph 1 Files

Nodes =
ID:1, Nodes:[15.5, 5.0], NodeInLinesID:['0', '1', '4']
ID:0, Nodes:[10.0, 0.0], NodeInLinesID:['0', '3']
ID:3, Nodes:[45.0, 0.0], NodeInLinesID:['2', '4']
ID:2, Nodes:[23.0, 5.0], NodeInLinesID:['1', '2', '3']

Lines =
ID:1, NodeStart:1, NodeEnd:2
ID:0, NodeStart:0, NodeEnd:1

Michelle Song
Chen Ren
Hw 4
ID:3, NodeStart:0, NodeEnd:2
ID:2, NodeStart:2, NodeEnd:3
ID:4, NodeStart:1, NodeEnd:3

The length from nodeID 1 to nodeID 0 is 7.43, passing the lineID['0']
The length from nodeID 1 to nodeID 0 is 21.43, passing the lineID['1', '3']
The length from nodeID 1 to nodeID 0 is 66.41, passing the lineID['4', '2', '3']
The shortest path above is 7.43,which passes the nodesID ['1', '0'] and the linesID ['0']

The length from nodeID 1 to nodeID 3 is 43.92, passing the lineID['0', '3', '2']
The length from nodeID 1 to nodeID 3 is 30.06, passing the lineID['1', '2']
The length from nodeID 1 to nodeID 3 is 29.92, passing the lineID['4']
The shortest path above is 29.92,which passes the nodesID ['1', '3'] and the linesID ['4']

The length from nodeID 1 to nodeID 2 is 21.36, passing the lineID['0', '3']
The length from nodeID 1 to nodeID 2 is 7.5, passing the lineID['1']
The length from nodeID 1 to nodeID 2 is 52.48, passing the lineID['4', '2']
The shortest path above is 7.5,which passes the nodesID ['1', '2'] and the linesID ['1']

The length from nodeID 0 to nodeID 1 is 7.43, passing the lineID['0']
The length from nodeID 0 to nodeID 1 is 21.43, passing the lineID['3', '1']
The length from nodeID 0 to nodeID 1 is 66.41, passing the lineID['3', '2', '4']
The shortest path above is 7.43,which passes the nodesID ['0', '1'] and the linesID ['0']

The length from nodeID 0 to nodeID 3 is 37.49, passing the lineID['0', '1', '2']
The length from nodeID 0 to nodeID 3 is 37.35, passing the lineID['0', '4']
The length from nodeID 0 to nodeID 3 is 51.35, passing the lineID['3', '1', '4']
The length from nodeID 0 to nodeID 3 is 36.49, passing the lineID['3', '2']
The shortest path above is 36.49,which passes the nodesID ['0', '2', '3'] and the linesID ['3', '2']

The length from nodeID 0 to nodeID 2 is 14.93, passing the lineID['0', '1']
The length from nodeID 0 to nodeID 2 is 59.91, passing the lineID['0', '4', '2']
The length from nodeID 0 to nodeID 2 is 13.93, passing the lineID['3']
The shortest path above is 13.93,which passes the nodesID ['0', '2'] and the linesID ['3']

The length from nodeID 3 to nodeID 1 is 30.06, passing the lineID['2', '1']
The length from nodeID 3 to nodeID 1 is 43.92, passing the lineID['2', '3', '0']
The length from nodeID 3 to nodeID 1 is 29.92, passing the lineID['4']
The shortest path above is 29.92,which passes the nodesID ['3', '1'] and the linesID ['4']

Michelle Song
Chen Ren
Hw 4

The length from nodeID 3 to nodeID 0 is 37.49, passing the lineID['2', '1', '0']
The length from nodeID 3 to nodeID 0 is 36.49, passing the lineID['2', '3']
The length from nodeID 3 to nodeID 0 is 37.35, passing the lineID['4', '0']
The length from nodeID 3 to nodeID 0 is 51.35, passing the lineID['4', '1', '3']
The shortest path above is 36.49,which passes the nodesID ['3', '2', '0'] and the linesID ['2', '3']

The length from nodeID 3 to nodeID 2 is 22.56, passing the lineID['2']
The length from nodeID 3 to nodeID 2 is 51.28, passing the lineID['4', '0', '3']
The length from nodeID 3 to nodeID 2 is 37.42, passing the lineID['4', '1']
The shortest path above is 22.56,which passes the nodesID ['3', '2'] and the linesID ['2']

The length from nodeID 2 to nodeID 1 is 7.5, passing the lineID['1']
The length from nodeID 2 to nodeID 1 is 52.48, passing the lineID['2', '4']
The length from nodeID 2 to nodeID 1 is 21.36, passing the lineID['3', '0']
The shortest path above is 7.5,which passes the nodesID ['2', '1'] and the linesID ['1']

The length from nodeID 2 to nodeID 0 is 14.93, passing the lineID['1', '0']
The length from nodeID 2 to nodeID 0 is 59.91, passing the lineID['2', '4', '0']
The length from nodeID 2 to nodeID 0 is 13.93, passing the lineID['3']
The shortest path above is 13.93,which passes the nodesID ['2', '0'] and the linesID ['3']

The length from nodeID 2 to nodeID 3 is 37.42, passing the lineID['1', '4']
The length from nodeID 2 to nodeID 3 is 22.56, passing the lineID['2']
The length from nodeID 2 to nodeID 3 is 51.28, passing the lineID['3', '0', '4']
The shortest path above is 22.56,which passes the nodesID ['2', '3'] and the linesID ['2']

Graph 2 Files

Nodes =
ID:Oakland, Nodes:[15.4, 12.0], NodeInLinesID:['SWA1', 'SWA3']
ID:Miami, Nodes:[5.0, 5.5], NodeInLinesID:['UA3', 'JB1']
ID:Denver, Nodes:[25.0, 10.0], NodeInLinesID:['AK2', 'SWA1', 'UA1', 'UA2', 'UA3']
ID:New York, Nodes:[2.0, 14.0], NodeInLinesID:['UA1', 'JB1']
ID:San Francisco, Nodes:[15.5, 12.0], NodeInLinesID:['AK1', 'UA2']
ID:Seattle, Nodes:[16.0, 18.0], NodeInLinesID:['AK1', 'AK2', 'SWA3']

Lines =
ID:JB1, NodeStart:New York, NodeEnd:Miami
ID:UA1, NodeStart:Denver, NodeEnd:New York
ID:AK2, NodeStart:Seattle, NodeEnd:Denver

Michelle Song
Chen Ren
Hw 4
ID:AK1, NodeStart:Seattle, NodeEnd:San Francisco
ID:UA2, NodeStart:Denver, NodeEnd:San Francisco
ID:SWA1, NodeStart:Oakland, NodeEnd:Denver
ID:SWA3, NodeStart:Oakland, NodeEnd:Seattle
ID:UA3, NodeStart:Denver, NodeEnd:Miami

The length from nodeID Oakland to nodeID Miami is 42.17, passing the lineID['SWA1', 'UA1', 'JB1']
The length from nodeID Oakland to nodeID Miami is 30.31, passing the lineID['SWA1', 'UA3']
The length from nodeID Oakland to nodeID Miami is 54.12, passing the lineID['SWA3', 'AK1', 'UA2', 'UA1', 'JB1']
The length from nodeID Oakland to nodeID Miami is 42.26, passing the lineID['SWA3', 'AK1', 'UA2', 'UA3']
The length from nodeID Oakland to nodeID Miami is 50.43, passing the lineID['SWA3', 'AK2', 'UA1', 'JB1']
The length from nodeID Oakland to nodeID Miami is 38.57, passing the lineID['SWA3', 'AK2', 'UA3']
The shortest path above is 30.31,which passes the nodesID ['Oakland', 'Denver', 'Miami'] and the linesID ['SWA1', 'UA3']

The length from nodeID Oakland to nodeID Denver is 9.81, passing the lineID['SWA1']
The length from nodeID Oakland to nodeID Denver is 21.76, passing the lineID['SWA3', 'AK1', 'UA2']
The length from nodeID Oakland to nodeID Denver is 18.07, passing the lineID['SWA3', 'AK2']
The shortest path above is 9.81,which passes the nodesID ['Oakland', 'Denver'] and the linesID ['SWA1']

The length from nodeID Oakland to nodeID New York is 33.16, passing the lineID['SWA1', 'UA1']
The length from nodeID Oakland to nodeID New York is 39.32, passing the lineID['SWA1', 'UA3', 'JB1']
The length from nodeID Oakland to nodeID New York is 45.11, passing the lineID['SWA3', 'AK1', 'UA2', 'UA1']
The length from nodeID Oakland to nodeID New York is 51.27, passing the lineID['SWA3', 'AK1', 'UA2', 'UA3', 'JB1']
The length from nodeID Oakland to nodeID New York is 41.42, passing the lineID['SWA3', 'AK2', 'UA1']
The length from nodeID Oakland to nodeID New York is 47.58, passing the lineID['SWA3', 'AK2', 'UA3', 'JB1']
The shortest path above is 33.16,which passes the nodesID ['Oakland', 'Denver', 'New York'] and the linesID ['SWA1', 'UA1']

Michelle Song
Chen Ren
Hw 4
The length from nodeID Oakland to nodeID San Francisco is 27.87, passing the lineID['SWA1', 'AK2', 'AK1']
The length from nodeID Oakland to nodeID San Francisco is 19.52, passing the lineID['SWA1', 'UA2']
The length from nodeID Oakland to nodeID San Francisco is 12.05, passing the lineID['SWA3', 'AK1']
The length from nodeID Oakland to nodeID San Francisco is 27.78, passing the lineID['SWA3', 'AK2', 'UA2']
The shortest path above is 12.05,which passes the nodesID ['Oakland', 'Seattle', 'San Francisco'] and the linesID ['SWA3', 'AK1']

The length from nodeID Oakland to nodeID Seattle is 21.85, passing the lineID['SWA1', 'AK2']
The length from nodeID Oakland to nodeID Seattle is 25.54, passing the lineID['SWA1', 'UA2', 'AK1']
The length from nodeID Oakland to nodeID Seattle is 6.03, passing the lineID['SWA3']
The shortest path above is 6.03,which passes the nodesID ['Oakland', 'Seattle'] and the linesID ['SWA3']

The length from nodeID Miami to nodeID Oakland is 38.57, passing the lineID['UA3', 'AK2', 'SWA3']
The length from nodeID Miami to nodeID Oakland is 30.31, passing the lineID['UA3', 'SWA1']
The length from nodeID Miami to nodeID Oakland is 42.26, passing the lineID['UA3', 'UA2', 'AK1', 'SWA3']
The length from nodeID Miami to nodeID Oakland is 50.43, passing the lineID['JB1', 'UA1', 'AK2', 'SWA3']
The length from nodeID Miami to nodeID Oakland is 42.17, passing the lineID['JB1', 'UA1', 'SWA1']
The length from nodeID Miami to nodeID Oakland is 54.12, passing the lineID['JB1', 'UA1', 'UA2', 'AK1', 'SWA3']
The shortest path above is 30.31,which passes the nodesID ['Miami', 'Denver', 'Oakland'] and the linesID ['UA3', 'SWA1']

The length from nodeID Miami to nodeID Denver is 20.5, passing the lineID['UA3']
The length from nodeID Miami to nodeID Denver is 32.36, passing the lineID['JB1', 'UA1']
The shortest path above is 20.5,which passes the nodesID ['Miami', 'Denver'] and the linesID ['UA3']

The length from nodeID Miami to nodeID New York is 43.85, passing the lineID['UA3', 'UA1']
The length from nodeID Miami to nodeID New York is 9.01, passing the lineID['JB1']
The shortest path above is 9.01,which passes the nodesID ['Miami', 'New York'] and

Michelle Song
Chen Ren
Hw 4
the linesID ['JB1']

The length from nodeID Miami to nodeID San Francisco is 38.56, passing the lineID['UA3', 'AK2', 'AK1']
The length from nodeID Miami to nodeID San Francisco is 42.36, passing the lineID['UA3', 'SWA1', 'SWA3', 'AK1']
The length from nodeID Miami to nodeID San Francisco is 30.21, passing the lineID['UA3', 'UA2']
The length from nodeID Miami to nodeID San Francisco is 50.42, passing the lineID['JB1', 'UA1', 'AK2', 'AK1']
The length from nodeID Miami to nodeID San Francisco is 54.22, passing the lineID['JB1', 'UA1', 'SWA1', 'SWA3', 'AK1']
The length from nodeID Miami to nodeID San Francisco is 42.07, passing the lineID['JB1', 'UA1', 'UA2']
The shortest path above is 30.21,which passes the nodesID ['Miami', 'Denver', 'San Francisco'] and the linesID ['UA3', 'UA2']

The length from nodeID Miami to nodeID Seattle is 32.54, passing the lineID['UA3', 'AK2']
The length from nodeID Miami to nodeID Seattle is 36.34, passing the lineID['UA3', 'SWA1', 'SWA3']
The length from nodeID Miami to nodeID Seattle is 36.23, passing the lineID['UA3', 'UA2', 'AK1']
The length from nodeID Miami to nodeID Seattle is 44.4, passing the lineID['JB1', 'UA1', 'AK2']
The length from nodeID Miami to nodeID Seattle is 48.2, passing the lineID['JB1', 'UA1', 'SWA1', 'SWA3']
The length from nodeID Miami to nodeID Seattle is 48.09, passing the lineID['JB1', 'UA1', 'UA2', 'AK1']
The shortest path above is 32.54,which passes the nodesID ['Miami', 'Denver', 'Seattle'] and the linesID ['UA3', 'AK2']

The length from nodeID Denver to nodeID Oakland is 18.07, passing the lineID['AK2', 'SWA3']
The length from nodeID Denver to nodeID Oakland is 9.81, passing the lineID['SWA1']
The length from nodeID Denver to nodeID Oakland is 21.76, passing the lineID['UA2', 'AK1', 'SWA3']
The shortest path above is 9.81,which passes the nodesID ['Denver', 'Oakland'] and the linesID ['SWA1']

The length from nodeID Denver to nodeID Miami is 32.36, passing the lineID['UA1', 'JB1']
The length from nodeID Denver to nodeID Miami is 20.5, passing the lineID['UA3']
The shortest path above is 20.5,which passes the nodesID ['Denver', 'Miami'] and

Michelle Song
Chen Ren
Hw 4
the linesID ['UA3']

The length from nodeID Denver to nodeID New York is 23.35, passing the lineID['UA1']
The length from nodeID Denver to nodeID New York is 29.51, passing the lineID['UA3', 'JB1']
The shortest path above is 23.35,which passes the nodesID ['Denver', 'New York'] and the linesID ['UA1']

The length from nodeID Denver to nodeID San Francisco is 18.06, passing the lineID['AK2', 'AK1']
The length from nodeID Denver to nodeID San Francisco is 21.86, passing the lineID['SWA1', 'SWA3', 'AK1']
The length from nodeID Denver to nodeID San Francisco is 9.71, passing the lineID['UA2']
The shortest path above is 9.71,which passes the nodesID ['Denver', 'San Francisco'] and the linesID ['UA2']

The length from nodeID Denver to nodeID Seattle is 12.04, passing the lineID['AK2']
The length from nodeID Denver to nodeID Seattle is 15.84, passing the lineID['SWA1', 'SWA3']
The length from nodeID Denver to nodeID Seattle is 15.73, passing the lineID['UA2', 'AK1']
The shortest path above is 12.04,which passes the nodesID ['Denver', 'Seattle'] and the linesID ['AK2']

The length from nodeID New York to nodeID Oakland is 41.42, passing the lineID['UA1', 'AK2', 'SWA3']
The length from nodeID New York to nodeID Oakland is 33.16, passing the lineID['UA1', 'SWA1']
The length from nodeID New York to nodeID Oakland is 45.11, passing the lineID['UA1', 'UA2', 'AK1', 'SWA3']
The length from nodeID New York to nodeID Oakland is 47.58, passing the lineID['JB1', 'UA3', 'AK2', 'SWA3']
The length from nodeID New York to nodeID Oakland is 39.32, passing the lineID['JB1', 'UA3', 'SWA1']
The length from nodeID New York to nodeID Oakland is 51.27, passing the lineID['JB1', 'UA3', 'UA2', 'AK1', 'SWA3']
The shortest path above is 33.16,which passes the nodesID ['New York', 'Denver', 'Oakland'] and the linesID ['UA1', 'SWA1']

The length from nodeID New York to nodeID Miami is 43.85, passing the lineID['UA1', 'UA3']
The length from nodeID New York to nodeID Miami is 9.01, passing the lineID['JB1']
The shortest path above is 9.01,which passes the nodesID ['New York', 'Miami'] and

Michelle Song
Chen Ren
Hw 4
the linesID ['JB1']

The length from nodeID New York to nodeID Denver is 23.35, passing the lineID['UA1']
The length from nodeID New York to nodeID Denver is 29.51, passing the lineID['JB1', 'UA3']
The shortest path above is 23.35,which passes the nodesID ['New York', 'Denver'] and the linesID ['UA1']

The length from nodeID New York to nodeID San Francisco is 41.41, passing the lineID['UA1', 'AK2', 'AK1']
The length from nodeID New York to nodeID San Francisco is 45.21, passing the lineID['UA1', 'SWA1', 'SWA3', 'AK1']
The length from nodeID New York to nodeID San Francisco is 33.06, passing the lineID['UA1', 'UA2']
The length from nodeID New York to nodeID San Francisco is 47.57, passing the lineID['JB1', 'UA3', 'AK2', 'AK1']
The length from nodeID New York to nodeID San Francisco is 51.37, passing the lineID['JB1', 'UA3', 'SWA1', 'SWA3', 'AK1']
The length from nodeID New York to nodeID San Francisco is 39.22, passing the lineID['JB1', 'UA3', 'UA2']
The shortest path above is 33.06,which passes the nodesID ['New York', 'Denver', 'San Francisco'] and the linesID ['UA1', 'UA2']

The length from nodeID New York to nodeID Seattle is 35.39, passing the lineID['UA1', 'AK2']
The length from nodeID New York to nodeID Seattle is 39.19, passing the lineID['UA1', 'SWA1', 'SWA3']
The length from nodeID New York to nodeID Seattle is 39.08, passing the lineID['UA1', 'UA2', 'AK1']
The length from nodeID New York to nodeID Seattle is 41.55, passing the lineID['JB1', 'UA3', 'AK2']
The length from nodeID New York to nodeID Seattle is 45.35, passing the lineID['JB1', 'UA3', 'SWA1', 'SWA3']
The length from nodeID New York to nodeID Seattle is 45.24, passing the lineID['JB1', 'UA3', 'UA2', 'AK1']
The shortest path above is 35.39,which passes the nodesID ['New York', 'Denver', 'Seattle'] and the linesID ['UA1', 'AK2']

The length from nodeID San Francisco to nodeID Oakland is 27.87, passing the lineID['AK1', 'AK2', 'SWA1']
The length from nodeID San Francisco to nodeID Oakland is 12.05, passing the lineID['AK1', 'SWA3']
The length from nodeID San Francisco to nodeID Oakland is 27.78, passing the lineID['UA2', 'AK2', 'SWA3']

Michelle Song
Chen Ren
Hw 4
The length from nodeID San Francisco to nodeID Oakland is 19.52, passing the lineID['UA2', 'SWA1']
The shortest path above is 12.05,which passes the nodesID ['San Francisco', 'Seattle', 'Oakland'] and the linesID ['AK1', 'SWA3']

The length from nodeID San Francisco to nodeID Miami is 50.42, passing the lineID['AK1', 'AK2', 'UA1', 'JB1']
The length from nodeID San Francisco to nodeID Miami is 38.56, passing the lineID['AK1', 'AK2', 'UA3']
The length from nodeID San Francisco to nodeID Miami is 54.22, passing the lineID['AK1', 'SWA3', 'SWA1', 'UA1', 'JB1']
The length from nodeID San Francisco to nodeID Miami is 42.36, passing the lineID['AK1', 'SWA3', 'SWA1', 'UA3']
The length from nodeID San Francisco to nodeID Miami is 42.07, passing the lineID['UA2', 'UA1', 'JB1']
The length from nodeID San Francisco to nodeID Miami is 30.21, passing the lineID['UA2', 'UA3']
The shortest path above is 30.21,which passes the nodesID ['San Francisco', 'Denver', 'Miami'] and the linesID ['UA2', 'UA3']

The length from nodeID San Francisco to nodeID Denver is 18.06, passing the lineID['AK1', 'AK2']
The length from nodeID San Francisco to nodeID Denver is 21.86, passing the lineID['AK1', 'SWA3', 'SWA1']
The length from nodeID San Francisco to nodeID Denver is 9.71, passing the lineID['UA2']
The shortest path above is 9.71,which passes the nodesID ['San Francisco', 'Denver'] and the linesID ['UA2']

The length from nodeID San Francisco to nodeID New York is 41.41, passing the lineID['AK1', 'AK2', 'UA1']
The length from nodeID San Francisco to nodeID New York is 47.57, passing the lineID['AK1', 'AK2', 'UA3', 'JB1']
The length from nodeID San Francisco to nodeID New York is 45.21, passing the lineID['AK1', 'SWA3', 'SWA1', 'UA1']
The length from nodeID San Francisco to nodeID New York is 51.37, passing the lineID['AK1', 'SWA3', 'SWA1', 'UA3', 'JB1']
The length from nodeID San Francisco to nodeID New York is 33.06, passing the lineID['UA2', 'UA1']
The length from nodeID San Francisco to nodeID New York is 39.22, passing the lineID['UA2', 'UA3', 'JB1']
The shortest path above is 33.06,which passes the nodesID ['San Francisco', 'Denver', 'New York'] and the linesID ['UA2', 'UA1']

The length from nodeID San Francisco to nodeID Seattle is 6.02, passing the

Michelle Song
Chen Ren
Hw 4
lineID['AK1']
The length from nodeID San Francisco to nodeID Seattle is 21.75, passing the lineID['UA2', 'AK2']
The length from nodeID San Francisco to nodeID Seattle is 25.55, passing the lineID['UA2', 'SWA1', 'SWA3']
The shortest path above is 6.02,which passes the nodesID ['San Francisco', 'Seattle'] and the linesID ['AK1']

The length from nodeID Seattle to nodeID Oakland is 25.54, passing the lineID['AK1', 'UA2', 'SWA1']
The length from nodeID Seattle to nodeID Oakland is 21.85, passing the lineID['AK2', 'SWA1']
The length from nodeID Seattle to nodeID Oakland is 6.03, passing the lineID['SWA3']
The shortest path above is 6.03,which passes the nodesID ['Seattle', 'Oakland'] and the linesID ['SWA3']

The length from nodeID Seattle to nodeID Miami is 48.09, passing the lineID['AK1', 'UA2', 'UA1', 'JB1']
The length from nodeID Seattle to nodeID Miami is 36.23, passing the lineID['AK1', 'UA2', 'UA3']
The length from nodeID Seattle to nodeID Miami is 44.4, passing the lineID['AK2', 'UA1', 'JB1']
The length from nodeID Seattle to nodeID Miami is 32.54, passing the lineID['AK2', 'UA3']
The length from nodeID Seattle to nodeID Miami is 48.2, passing the lineID['SWA3', 'SWA1', 'UA1', 'JB1']
The length from nodeID Seattle to nodeID Miami is 36.34, passing the lineID['SWA3', 'SWA1', 'UA3']
The shortest path above is 32.54,which passes the nodesID ['Seattle', 'Denver', 'Miami'] and the linesID ['AK2', 'UA3']

The length from nodeID Seattle to nodeID Denver is 15.73, passing the lineID['AK1', 'UA2']
The length from nodeID Seattle to nodeID Denver is 12.04, passing the lineID['AK2']
The length from nodeID Seattle to nodeID Denver is 15.84, passing the lineID['SWA3', 'SWA1']
The shortest path above is 12.04,which passes the nodesID ['Seattle', 'Denver'] and the linesID ['AK2']

The length from nodeID Seattle to nodeID New York is 39.08, passing the lineID['AK1', 'UA2', 'UA1']
The length from nodeID Seattle to nodeID New York is 45.24, passing the lineID['AK1', 'UA2', 'UA3', 'JB1']
The length from nodeID Seattle to nodeID New York is 35.39, passing the

Michelle Song
Chen Ren
Hw 4
lineID['AK2', 'UA1']
The length from nodeID Seattle to nodeID New York is 41.55, passing the lineID['AK2', 'UA3', 'JB1']
The length from nodeID Seattle to nodeID New York is 39.19, passing the lineID['SWA3', 'SWA1', 'UA1']
The length from nodeID Seattle to nodeID New York is 45.35, passing the lineID['SWA3', 'SWA1', 'UA3', 'JB1']
The shortest path above is 35.39,which passes the nodesID ['Seattle', 'Denver', 'New York'] and the linesID ['AK2', 'UA1']

The length from nodeID Seattle to nodeID San Francisco is 6.02, passing the lineID['AK1']
The length from nodeID Seattle to nodeID San Francisco is 21.75, passing the lineID['AK2', 'UA2']
The length from nodeID Seattle to nodeID San Francisco is 25.55, passing the lineID['SWA3', 'SWA1', 'UA2']
The shortest path above is 6.02,which passes the nodesID ['Seattle', 'San Francisco'] and the linesID ['AK1']

Validation

We wanted to test if node coordinates could take negative values and if lines can go from node 1 to node 1 (test that code can handle this).

To do this, we changed the lines.txt and nodes.txt to

Lines.txt

0	0	1
1	1	1
2	2	3
3	0	2
4	1	3

Nodes.txt

0	-10	0.0
1	15.5	5.0
2	23.0	5.0
3	45.0	0.0

Output

From the output we can see that the code correctly read in negative node

Michelle Song
Chen Ren
Hw 4
coordinates (Node 0).
We can also see that the line from node 1 to node 1 does not appear in the output and does not affect any calculations.
The length/distance from node 0 to node 1 was also correctly calculated (as are other length calculations).

Nodes =
ID:1, Nodes:[15.5, 5.0], NodeInLinesID:['0', '1', '4']
ID:0, Nodes:[-10.0, 0.0], NodeInLinesID:['0', '3']
ID:3, Nodes:[45.0, 0.0], NodeInLinesID:['2', '4']
ID:2, Nodes:[23.0, 5.0], NodeInLinesID:['2', '3']

Lines =
ID:1, NodeStart:1, NodeEnd:1
ID:0, NodeStart:0, NodeEnd:1
ID:3, NodeStart:0, NodeEnd:2
ID:2, NodeStart:2, NodeEnd:3
ID:4, NodeStart:1, NodeEnd:3

The length from nodeID 1 to nodeID 0 is 25.99, passing the lineID['0']
The length from nodeID 1 to nodeID 0 is 85.86, passing the lineID['4', '2', '3']
The shortest path above is 25.99,which passes the nodesID ['1', '0'] and the linesID ['0']

The length from nodeID 1 to nodeID 3 is 81.93, passing the lineID['0', '3', '2']
The length from nodeID 1 to nodeID 3 is 29.92, passing the lineID['4']
The shortest path above is 29.92,which passes the nodesID ['1', '3'] and the linesID ['4']

The length from nodeID 1 to nodeID 2 is 59.37, passing the lineID['0', '3']
The length from nodeID 1 to nodeID 2 is 52.48, passing the lineID['4', '2']
The shortest path above is 52.48,which passes the nodesID ['1', '3', '2'] and the linesID ['4', '2']

The length from nodeID 0 to nodeID 1 is 25.99, passing the lineID['0']
The length from nodeID 0 to nodeID 1 is 85.86, passing the lineID['3', '2', '4']
The shortest path above is 25.99,which passes the nodesID ['0', '1'] and the linesID ['0']

The length from nodeID 0 to nodeID 3 is 55.91, passing the lineID['0', '4']
The length from nodeID 0 to nodeID 3 is 55.94, passing the lineID['3', '2']
The shortest path above is 55.91,which passes the nodesID ['0', '1', '3'] and the linesID ['0', '4']

The length from nodeID 0 to nodeID 2 is 78.47, passing the lineID['0', '4', '2']

Michelle Song
Chen Ren
Hw 4
The length from nodeID 0 to nodeID 2 is 33.38, passing the lineID['3']
The shortest path above is 33.38,which passes the nodesID ['0', '2'] and the linesID ['3']

The length from nodeID 3 to nodeID 1 is 81.93, passing the lineID['2', '3', '0']
The length from nodeID 3 to nodeID 1 is 29.92, passing the lineID['4']
The shortest path above is 29.92,which passes the nodesID ['3', '1'] and the linesID ['4']

The length from nodeID 3 to nodeID 0 is 55.94, passing the lineID['2', '3']
The length from nodeID 3 to nodeID 0 is 55.91, passing the lineID['4', '0']
The shortest path above is 55.91,which passes the nodesID ['3', '1', '0'] and the linesID ['4', '0']

The length from nodeID 3 to nodeID 2 is 22.56, passing the lineID['2']
The length from nodeID 3 to nodeID 2 is 89.29, passing the lineID['4', '0', '3']
The shortest path above is 22.56,which passes the nodesID ['3', '2'] and the linesID ['2']

The length from nodeID 2 to nodeID 1 is 52.48, passing the lineID['2', '4']
The length from nodeID 2 to nodeID 1 is 59.37, passing the lineID['3', '0']
The shortest path above is 52.48,which passes the nodesID ['2', '3', '1'] and the linesID ['2', '4']

The length from nodeID 2 to nodeID 0 is 78.47, passing the lineID['2', '4', '0']
The length from nodeID 2 to nodeID 0 is 33.38, passing the lineID['3']
The shortest path above is 33.38,which passes the nodesID ['2', '0'] and the linesID ['3']

The length from nodeID 2 to nodeID 3 is 22.56, passing the lineID['2']
The length from nodeID 2 to nodeID 3 is 89.29, passing the lineID['3', '0', '4']
The shortest path above is 22.56,which passes the nodesID ['2', '3'] and the linesID ['2']