CS112 Summer 2017: Assignment 4

Shortest Path

Due

Early **Friday** morning August 11, at **2:00 AM**

Shortest Path

This assignment is to write code for Dijkstra's shorted path algorithm.

Files to download

assig4.zip

Do NOT unzip this file. Instead, click on the Eclipse tab on Sakai, and follow the instructions in the section titled Importing a Zipped Project Into Eclipse, and import this zip file.

Representing the Graph

There are two representations of the graph we will be working with: The representation as a file and the representation in the program. (Note that the graph is undirected.)

In a File

A graph is represented by a file with the following format:

- The first line is a single number, the number of nodes in the graph
- Then there is a line for each vertex giving the name of the vertex. Each of these lines has a single word (i.e., no spaces or other whitespace).
- Then there is a line for every edge in the graph. Each of these lines has a name, one or more spaces, another name, one or more spaces, and a number. The names are the nodes this edge connects and the number is the weight of this edge.

E.g., the file

```
4
AA
BBB
Cat
Dog
```

AA BBB 3
Cat Dog 6
AA Cat 4

represents the graph

```
            3
  AA  ----------------- BB

   |

   | 4

   |
            6
 Cat ----------------- Dog
```

In the program

Classes Graph, Vertex and AdjacencyNode are used to represent the graph we are working on:

- A Graph object represents the graph. It has one field, adjLists, which is an array of Vertex objects, representing the vertices of the graph. In many places in the program, a vertex is represented by an int which an index into this array – adjLists[v] is the Vertex object that holds the name and adjacency list for the vertex represented by the int v.

  The constructor for Graph is already written. It takes a String which is a file name of a file in the format described above. It reads the file and initializes the Graph to be the graph represented by the file. The main method shows how to create a Graph object. The method indexForName converts a string name into an int swhich is an index into the Graph's adjLists array. All of Graph is already written except the method shortestPath. (See below.)

- A Vertex represents a single vertex of the graph, with fields name (a String) and adjList (the first AdjacencyNode in the adjacency list of the vertex). The class Vertex is already written.

- An AdjacencyNode is a node in an adjacency list. It represents an edge. It has fields weight (an int) which is the weight of the edge, and vertexNum which is the index into the graph's adjList array that represents the vertex at the other end of the edge. The class AdjacencyNode is already written.

Implementation

Your task it to complete the method shortestPath in class Graph, using Dijkstra's algorithm which we went over in class. The data structure we used, i.e., the table giving, for each node, the status, link, and distance for that node, is to be represented by an array of SPRecord objects. A vertex represented by the int j has its SPRecord object at index j of this array. (The array is to be declared in the shortestPath method.) ShortestPath takes two strings as arguments, the names of two vertices. The method shortestPath is to compute the length of the shortest path between these two vertices. The method indexForName can be used to convert a name to the corresponding index into adjLists and into the array of SPRecords.

Sone rules:

- You may not change any class (any .java file) besides Graph. (This is the only file you will hand in.)

- You must finish the definition of shortestPath in Graph.java

- You may add other methods to class Graph as long as they are private.

Handing in

This assignment is due Friday morning August 11 at **2:00 AM**. Attach your Graph.java to this assignment.