Simple Graph Algorithms

Write a Python program to answer a series of questions about a graph that is given in the file GraphData.txt. The structure of GraphData.txt is that each line is given in the form

Vertex1,Vertex2

To simplify things, the vertices are given as integers, but not necessarily consecutive integers. Your Python program should start by reading in all the data in GraphData.txt and finish by producing a text file called GraphDataOut.txt which is structured as described below. You should include your Python program as a .py file and GraphDataOut.txt in your submitted ZIP le. You do not need to include GraphData.txt in the ZIP le. If you can't do some parts of this problem place the text I could not do this part in the appropriate place in the output file.

(a) (2 points) The first line should read "The number of vertices in the graph is NumVert." where NumVert is the number of distinct vertices in the graph.

(b) (2 points) The second line should read "The number of edges in the graph is NumEdge." where NumEdge is the number of distinct edges in the graph. Note that GraphData.txt might contain duplicate edges.

(c) (1 point) The third line should read "Below is the adjacency list for this graph with the vertices sorted."

(d) (3 points) The next NumVert lines of the text file should contain the adjacency list for the graph with one list element per line in the form

Vertex,Neighbor1,Neighbor2,...

where all neighbors of the vertex are given. The lines of the adjacency list must be sorted by initial vertex and the list of neighbors must also be sorted. Make the lines as long as necessary to include all neighbors.

(e) (4 points) Following the adjacency list, the next line of the file should read "The number of connected components of this graph is NumComp." where NumComp is the number of connected components.

(f) (1 point) The next line should read "The number of connected components of the graph that have an Euler path is NumEulerPath." where NumEulerPath is the number of components that have an Euler Path.

(g) (3 points) The next section of the file should list all the Eulerian paths in the graph with the following format. Before each Eulerian path you should have a line that reads "The following NumEdges lines list the edges for the Eulerian path for Component NumEPComp." where NumEPComp is the number of the component that you are listing and NumEdges is the number of edges in that component. The lines must be listed in the format that describes a path, in other words if one line looks like

vertex1,vertex2

then the next line should look like

vertex2,vertex3

The components may be listed in any order. The first vertex on the first line should be the starting point of your Euler path and the last vertex on the last line should be the finishing point of your Euler path.

(h) (1 point) The next line should read "The number of connected components of the graph that have an Euler circuit is NumEulerCirc." where NumEulerCirc is the number of components that have an Euler circuit.

(i) (3 points) The next section of the file should list all the Eulerian circuits in the graph with the following format. Before each Eulerian circuit you should have a line that reads "The following NumEdges lines list the edges for the Eulerian path for Component NumECComp." where NumECComp is the number of the

components that you are listing and NumEdges is the number of edges in that component. The lines must be listed in the format that describes a circuit, in other words if one line looks like

vertex1,vertex2

then the next line should look like

vertex2,vertex3

The components may be listed in any order. The first vertex on the first line should be the starting point of your Euler circuit and the last vertex on the last line should be the finishing point of your Euler circuit, i.e., these two should be the same vertex.