CS 6301.001. Implementation of data structures and algorithms
Long Project 2: Euler tours in directed graphs

Ver 1.0: Initial description (Wed, Sep 24).
Due: 11:59 PM, Wed, Oct 9 (1st deadline), Sun, Oct13 (2nd deadline).

Max excellence credits: 1.0.

- **Submit before the first deadline to be eligible for excellence credit.**
- Submission procedure is same as the same as that of prior projects.
- For each group, only its last submission is kept and earlier submissions are discarded.
- Your code must be of good quality, and pass all test cases within time limits to earn excellence credits.

Project Description

A directed graph is called Eulerian if it is strongly connected, and the number of incoming edges at each node is equal to the number of outgoing edges from it. It is known that such a graph always has a tour (a cycle that may not be simple) that goes through every edge of the graph exactly once. Such a tour (sometimes called a circuit) is called an Euler tour. In this project, write code that finds an Euler tour in a given directed graph using one of the algorithms described in class. Your program will be tested with large sized graphs, say, 200,000 nodes. You need to ensure that your implementation is efficient to handle large input size. If your code runs slowly, you may not get excellence credit.

Euler.java

This is the main class that you need to complete. Some skeleton code is provided. You can add additional methods as needed. Do not change the signatures of any of the methods called by the driver

Driver code for the project is in main(). It works as follows. If a command line parameter is there, it is treated as the name of a file that has the input data. Otherwise the program reads its input from the console. If the command line has a second parameter (optional), it is taken as the start vertex of the tour. An optional third parameter can be used to suppress the output, or to print verbose output. Suppression of output is convenient when the graph is huge, and printing the (intermediate) output consumes a lot of time, skewing the actual running time of the program to find an Euler tour.

Input/Output

The input is given in the format expected by Graph.readDirectedGraph(). First two integers read are |V| and |E|. The next |E| lines of input have 3 integers each: u, v, and w, indicating an edge (u,v) with weight w. In this problem, the weights of edges are irrelevant. If the graph is not

Eulerian, the method isEulerian() in Euler.java will output the message: Graph is not Eulerian, and provide a justification for that claim with something like "Graph is not strongly connected" or "inDegree = 5, outDegree = 3 at Vertex 37". Otherwise, the graph is Eulerian, and the program should output a tour, by printing a sequence of edges, starting at the chosen start vertex, visiting every edge of G exactly once.