CSCI 305 Concepts of Programming Languages

Programming Lab 2 — Reconstructing Montana's Road Network

Python:

For this lab you will need to learn and use Python. (You have seen a few examples in class, but you do need to learn more on Python to finish this lab.) Python is installed on `esus`, or most Linux systems. Type `python -v` to check your version. If you intend to implement a graphical interface, you might need to download and use the latest version of Python, say from `https://www.python.org`.

Dataset:

This lab will process a dataset containing the direct connection distances between cities in Montana. (Each connection does not necessarily exist in our current road network.) Assume that some devastating earthquakes occurred and the direct road connections between some cities were broken, say between Bozeman and Billings. Assume also that all other tools like GPS do not work anymore and all we have is this file:

`http://www.cs.montana.edu/bhz/classes/spring-2016/csci305/LAB2/city1.txt`

You need to first import this file and build a data structure to store it. Then you need to build a tool to facilitate the reconstruction of Montana's Road Network. Your Python code must support the following queries:

Task 1. Number of cities directly connected to a query city.

For instance, if you input: `Bozeman`, the result should be 23.

Task 2. Given two query cities, return YES/NO for whether there is a direct connection (edge) between them.

For instance, if you input: `Bozeman, Billings`, the result should be NO.

Task 3. Given two query cities and an integer d, return YES/NO for whether there is a k-hop connection, $k \leq d$, between them; if YES, print one solution out, to gether with the total distance of the d hops.

We say that $\langle v_1, v_2, ..., v_{d+1} \rangle$ forms a d-hop connection between v_1 and v_{d+1} if (v_i, v_{i+1}) is a direct link for $i = 1, ..., d$ (i.e., between two cities v_i and v_{i+1} there is a non-zero distance in `city.txt`).

Task 4. Given two query cities, return YES/NO for whether there is a connection (not necessarily direct) between them; if YES, print one solution out, together with the actual total distance of the connection.

As the dataset is not huge compared with Lab 1, you need to come up ways to test the correctness of your code.

Troubleshooting

This lab requires an independent study of the Python language. Use any web tutorials and resources to learn Python. Given the size of the class, we will not be able to debug your code for you. Please do not send panicked emails requesting us fix your bug for you — same as when you work for a company. Do allow yourself plenty of time, and use patience, perseverance, and the internet to debug your code. *On the other hand, if you do need help for data structures and general ideas, you are welcome to contact us.*

Submission

Each student will complete and submit this assignment individually or in a 2-person team. Comment your program properly, intelligent comments and a clean, readable formatting of your code account for 20% of your grade for this lab.

Save the final version of your program as `lastname_firstname.lab2.py`. Submit the final version of your program. *At the header, please supply the version of Python that your code was run; and, if you ran your code unconventionally, please supply a brief instruction on how to run your code.*

Team members will submit identical code. But each student must submit individually and at the beginning of your code you must state clearly your name and your partner's name (if any). Submit your file to the Lab2 dropbox folder on D2L. Note that late submissions will not be accepted.

DEADLINE: March, 17, 2016; 11:30pm.