

1 Problem

We want to create and display a map of any TIGER county displaying ONLY the TIGER Lines. The map supports zooming and panning. The data must be stored in a Postgres database. So there are 3 parts in this homework:

- (a) Write a program called **TigerLoader** to enter the appropriate TIGER data into a Postgres database.
- (b) Write a GUI program called **MapDisplay** to access and display the data in the TIGER database.

GUI Specifications. Implement this program as a resizable window that contains two panels called Viewing Panel and a Control Panel below it. You must outline the borders of these two panels. Your map is displayed in the Viewing Panel. The height of the Control Panel is fixed, but enough to display the various controls. In particular, we want two buttons for zooming in and out. Panning is achieved by mouse drag. You must take care that the image does not "disappear" when zooming (this amounts to ensuring that the center of your zoom is centered at the current mouse position).

JDBC, Postgres and TIGER files You need to write a Java program program to load any TIGER file into the database. This program must be flexible enough to read ANY TIGER file. For this purpose, we define the concept of a format file, as explained in Lecture 9. You may modify the examples (RawTigerReader, SReader) found in Lecture 9. For this homework, we only need two files for this homework: RT1 and RT2. So you must write the appropriate format files for RT1 and RT2. You will also want to use an R-Tree index on the bounding box of Tiger lines, to support rapid retrieval of these lines.

Graphics. How should you draw on the Viewing Panel? We suggest that you do this in two steps: create a BufferedImage called bI in which you draw your map. Then you display bI on your Viewing Panel. The advantage of this is that (1) you avoid flickering and (2) you can manipulate this image via rescaling or translation to speed up some operations. Here is how you get a BufferedImage bI, and then get an associated graphics object g2:

```
BufferedImage bI = new BufferedImage(width, height, TYPE_INT_ARGB);  
Graphics2D g2 = bI.createGraphics();
```

We recommend that (width, height) here be twice your Viewing Panel dimensions. Using g2, you can draw your TIGER lines onto the BufferedImage. Finally, to display bI on the Viewing Panel, you use the `drawImage(...)` methods of graphics (see accompanying notes).