CENG 112 – DATA STRUCTURES
Homework 2

March 18, 2016

Due Date: April 1st, 2016

Programming Assignment 1.1 Image ADT

We would like to implement an ADT that can hold and manipulate grayscale (as opposed to color) image data. Images on a digital computer are stored as arrays of bytes and each byte stores the grayscale intensity level of a pixel between 0 and 255. The top-left corner of the image is at `(0, 0)`, the top-most row of the image corresponds to `y` = 0, so `y` grows downwards and `x` towards right.

Implement a public class named `Image` that has the following public API:

- A constructor `public Image(int width, int height)` that creates an image of size `width` pixels wide and `height` pixels high.
- A method `public void setPixel(int x, int y, byte intensity)` that sets the pixel at coordinates `(x, y)` to the value `intensity`.
- A method `public void savePGM(String filename)` that saves the image in the ASCII or binary PGM format. You can choose and support one of the formats.
- A method `public void loadPGM(String filename)` that loads the image from the ASCII or binary PGM format. You can choose and support one of the formats.

To implement the ADT you need to choose a representation for pixel data. For this exercise, we will use a single private byte array `data` to store the intensity levels of each pixel. `data[y*width+x]` will store the intensity of pixel at coordinates `(x, y)`. You also need private fields to keep track of `width` and `height`.

To save and load the image data use the `In` and `Out` classes from the book's library. Use the online resources to learn about the PGM format.

Programming Assignment 1.2 Testing the Image ADT

Write a program named `ImageTester` that creates a checkerboard image of size `(w, h)` with square size `s` and saves it as `checkerboard.pgm` in the current directory.

A checkerboard image contains alternating black and white squares each of the same size. You can start with either a black or white square on the top-left corner of the image.

The parameters `w`, `h`, and `s` should be command line arguments so that you can run the program as `java ImageTester 320 340 20` for example.

Hint: Start by writing a program that creates a fixed size image, for example a 80x80 image with squares of size 20 by using loops with fixed ranges to construct the image. Then generalize using the command-line parameters.

Programming Assignment 1.3 Image ADT implementation with arrays of rows

In a new class `ImageR`, reimplement the image ADT using an array of arrays of bytes `byte[][] data` so that `data[0]` is a byte array containing row 0 and pixel at `(x, y)` is stored in `data[y][x]`. Modify the `ImageTester` class to use this new implementation. How much did you need to change?