Spring ’11 CIS 212 Homework 3 – 120/100 points possible – Due Monday, 4-18, 11:59 PM

The goal of this assignment is to provide experience with writing a simple sorting algorithm and further experience with creating a widget-based GUI. The exercises from the book are indented to provide experience with analyzing algorithmic complexity and determining the order-of-magnitude cost of an algorithm with respect to its input(s).


3. [10] Create a new Eclipse Java project named SortingUI. Add your Password and PasswordList classes from the previous assignment, but have PasswordList now extend java.util.Observable. You can remove the getPasswordBySequentialSearch() and getPasswordByBinarySearch() methods from your PasswordList class if you’d like (we won’t be using them in this assignment).

4. [30] Add a public selectionSort() method to your PasswordList class which takes no parameters. The method should implement selection sort by sorting the parsed data. After each iteration of the outer sorting loop, use the appropriate methods in Observable to flag the instance as changed and notify its observers (see the java.util.Observable documentation). Add a public getProgress() method to your PasswordList which takes no parameters and returns an int between 0 and 100 (inclusive) indicating the completion percentage of the sorting process with respect to progress through the outer loop. For example, if your outer sorting loop is currently on index 500 of 1000 elements, getProgress() should return 50. getProgress() should return 100 when sorting is complete.

5. [10] Add a SelectionSortThread class to your project. Have SelectionSortThread extend java.lang.Thread. The SelectionSortThread constructor should take a PasswordList as a parameter and the class should call selectionSort() on the PasswordList instance when the thread is started (see the java.lang.Thread documentation). Starting the selection sort from a separate thread will allow the GUI to continue to update during the sorting process (we’ll discuss in class).

6. [30] Add a MainFrame class to your project with a static main() method which creates a new MainFrame instance. Have MainFrame extend javax.swing.JFrame and implement java.util.Observer. Title the frame “SortingUI”. The MainFrame constructor should create an instance of PasswordList by passing the filename “unsorted.txt” and should register itself as an observer of the PasswordList instance (again, see the java.util.Observable documentation). The unsorted.txt file can be downloaded here or from the course announcements page. The MainFrame constructor should also create a javax.swing.JButton which displays the text “Start”
and a javax.swing.JProgressBar. Use BorderLayout to arrange the JProgressbar in the center of the frame and the JButton in the south of the frame. Add a java.awt.event.ActionListener to the JButton which creates and starts a new SelectionSortThread when the button is clicked. Upon receiving updates from the PasswordList (see java.util.Observer documentation), yourMainFrame class should get the current progress of the sorting algorithm using the PasswordList’s getProgress() method and use the returned value to set the progress of the JProgressBar. Your GUI should look like this:

![GUI Image]

7. [+20] Use the java.awt.Graphics class to create your own progress bar and have your MainFrame class use this progress bar in place of a JProgressBar instance.

Zip the SortingUI folder in your Eclipse workspace directory along with your problem-set solution document. Name your zip file <your full name>Homework3.zip and upload to Blackboard.