Histogram - Grade Distribution

1 Introduction

This assignment will have you working with 1-D arrays and console/file I/O.

2 Problem Description

You are a Georgia Tech professor teaching a class. It is the end of the semester and you wish to see how your students performed, so you write a Java program that will create a histogram of the grade distribution. You want this histogram program to be able to give you a very detailed view or a very broad view of the grade distribution. To accomplish this, it asks the user how many bins the grades should be split into.

3 Solution Description

We have provided you with a grade text file that has one integer number per line representing a student's grade in the class. These numbers are not sorted but they are bound between 0 and 100 (inclusive). Using an array, you must count the frequency of each grade value and print it to the standard output as a horizontal histogram. You must also label the range of each histogram bar and allow the user to indicate what size interval they would like the histogram to be made with.

Running the program should look something like this:

> Note: `$` is the command prompt on Unix. On Windows it will look something like `C:`
> >.

```
$ java GradeHistogram grades.txt
Grades loaded!
What bucket size would you like?
>>> 10

100 - 91 | [][][][][][][][][][][][]
 90 - 81 | [][][][][][][][][][][][][][][][][][][][][][][][][][]
 80 - 71 | [][][][][][][][][][][][][][]
 70 - 61 | [][][][][][][][][][][][][][][][][]
 60 - 51 | [][][][][][][]
 50 - 41 | [][][][]
 40 - 31 | [][][][][][]
 30 - 21 | [][]
 20 - 11 |
 10 -  1 | []
  0 -  0 |

$ java GradeHistogram grades.txt
Grades loaded!
What bucket size would you like?
>>> 5

100 - 96 | [][][][][][]
 95 - 91 | [][][][][][]
 90 - 86 | [][][][][][][][][][][][][][][][][]
 85 - 81 | [][][][][][][][][]
 80 - 76 | [][][][][][][][][][][][]
 75 - 71 | [][]
 70 - 66 | [][][][][][][][][][]
 65 - 61 | [][][][][][][]
 60 - 56 | [][][][][]
 55 - 51 | [][]
 50 - 46 | [][][]
 45 - 41 | []
 40 - 36 | [][][]
 35 - 31 | [][][]
 30 - 26 | []
 25 - 21 | []
 20 - 16 |
 15 - 11 |
 10 -  6 | []
  5 -  1 |
  0 -  0 |
```

The pipe characters should be aligned and your program must not exclude any subrange between 0 and 100.

4 Checkstyle

Review the CS 1331 Code Conventions and download the Checkstyle jar file and the configuration file to the directory that contains your Java source files. Run Checkstyle on your code like this (in the directory containing all your Java source files):

```
$ java -jar checkstyle-5.6-all.jar -c cs1331-checkstyle.xml *.java
Starting audit...
Audit done.
```

The message above means there were no Checkstyle errors. You can easily count the errors by piping the output of Checkstyle through `grep`.

```
$ java -jar checkstyle-5.6-all.jar -c cs1331-checkstyle.xml *.java | grep -cEv "(Starting
    audit...|Audit done)"
0
$
```

The `-c` option tells `grep` to count matching lines instead of printing them, `E` means use `egrep` (extended `grep`) syntax, and `v` means invert the match. Here we use an inverted match to discard the two non-error lines of Checkstyle's output. If you have `egrep` you could leave off the `-E` and just use `egrep`.

```
$ java -jar checkstyle-5.6-all.jar -c cs1331-checkstyle.xml *.java | egrep -cv "(Starting
    audit...|Audit done)"
0
$
```

The Java source files we provide contain no Checkstyle errors. You are responsible for any Checkstyle errors you introduce when modifying these files. For this assignment, there will be a maximum of 10 points lost due to Checkstyle errors. In future homeworks we will be increasing this cap, so get into the habit of fixing these style errors early!

5 Tips

- You may assume that you always get valid input.
- You may assume the text file has valid numbers.
- 101 is a prime number.
- An array is a fixed size data structure; you need to know ahead of time how big it needs to be. How do we do this?
- You can give interpretations to the indices and contents of an array to arrive at creative solutions to problems. Code smart, not hard.
- Creating a `Scanner` object with a file will throw a checked exception. Don't worry about what this means — for now, just append `throws Exception` to the end of the `main` method signature wherein the file is opened.

6 Turn-in Procedure

Submit all of the Java source files you created to T-Square **in addition to the text file we provided you**. Do not submit any compiled bytecode (`.class` files), the Checkstyle jar file, or the `cs1331-checkstyle.xml` file. When you're ready, double-check that you have submitted and not just saved a draft.

7 Verify the Success of Your Submission to T-Square

Practice safe submission! Verify that your HW files were truly submitted correctly, the upload was successful, and that the files compile and run. It is solely your responsibility to turn in your homework and practice this safe submission safeguard.

1. After uploading the files to T-Square you should receive an email from T-Square listing the names of the files that were uploaded and received. If you do not get the confirmation email almost immediately, something is wrong with your HW submission and/or your email. Even receiving the email does not guarantee that you turned in exactly what you intended.

2. After submitting the files to T-Square, return to the Assignment menu option and this homework. It should show the submitted files.

3. Download copies of your submitted files from the T-Square Assignment page placing them in a new folder.

4. Recompile and test those exact files.

5. This helps guard against a few things.

   (a) It helps insure that you turn in the correct files.

   (b) It helps you realize if you omit a file or files.[1] (If you do discover that you omitted a file, submit all of your files again, not just the missing one.)

   (c) Helps find last minute causes of files not compiling and/or running.

[1] Missing files will not be given any credit, and non-compiling homework solutions will receive few to zero points. Also recall that late homework will not be accepted regardless of excuse. Treat the due date with respect. The real due date is midnight Wednesday. Do not wait until the last minute!