Assignment 1: *Design*

Sriramya Prayaga

January 13, 2022

This assignment asks us to write a bash script that creates various graphs with Gnuplot using data obtained from the Collatz sequence (provided by the given file collatz.c). The first plot graphs the length of the Collatz sequences of the numbers 2 to 10,000. The second graph plots the maximum value of the Collatz sequences of the numbers 2 to 10,000. The third graph reveals the frequency of each Collatz sequence length for the numbers from 2 to 10,000.

While doing this assignment, as I became interested in seeing what the lengths of the sequences would be if I ran the given collatz program randomly, I also plotted the value of 99 random runs against their lengths using dots.

1 Pseudocode

(graph 1 pseudocode below)

if the data file for this graph already exists, delete it

for each i in the range of 2 to 10,000, print i and run the collatz.c file (using i as the input to the program) to count the number of output lines each i creates (length of sequence) and append i and this value to a data file.

Then with Gnuplot:

- set the terminal to pdf format
- set the output to the desired pdf file
- set the title of the graph to "Collatz Sequence Lengths"
- set the label of the x-axis to "x"
- set the label of the y-axis to "length"
- set the key off
- plot the values in the data file with dots

(graph 2 pseudocode below)

if the data file for this graph already exists, delete it

for each i in range 2 to 10,000, print i and run the collatz.c file (using i as the input), then reversely sort the numerical output values of each i to append i and its first sorted value to a data file.

Then with Gnuplot:

- set the terminal to pdf format
- set the output to the desired pdf file
- set the title of the graph to "Maximum Collatz Sequence Value"
- set the label of the x-axis to "n"
- set the label of the y-axis to "value"
- set the key off
- plot the values in the data file using the default x-range and specified y-range [0:100000] with dots

(graph 3 pseudocode below)
if the data file for this graph already exists, delete it
get the values of the second column of the data file of the first graph, sort those values, compute the frequency of each length, and redirect these new values to a new data file
Then with Gnuplot:

- set the terminal to pdf format
- set the output to the desired pdf file
- set the title of the graph to "Maximum Collatz Sequence Value"
- set the label of the x-axis to "length"
- set the label of the y-axis to "frequency"
- set the key off
- set xtics values with 0,25,225
- plot the values in the latter data file using the default y-range and specified x-range [0:225] and treating the second column in the latter data file as x, and first column as y with dots

(graph 4 pseudocode below)
if the data file for this graph already exists, delete it
for each i in the range of 2 to 100, print i and run the collatz.c file (randomly) to count the number of output lines each i creates (length of the sequence), sleep for 1 second, and append i and its corresponding length to a data file.
Then with Gnuplot:

- set the terminal to pdf format
- set the output to the desired pdf file
- set the title of the graph to "Lengths of 100 Random Collatz Sequences (with different seeds)"
- set the label of the x-axis to "random sequence index number (max index is 100)"

- set the label of the y-axis to ”length”
- set the key off
- plot the values in the data file with dots

2 Explanation of Pseudocode

The first two plots and the last plot are constructed from data created by for-loops, which can be observed above in the pseudocode. My code also deletes the data file for each plot if it already exists before the data is generated for the graphs. I did this because the for-loops would append more data to the existing file if I didn't delete the file (this would mess up the plotting.)

2.1 Explanation of the First Graph

The first graph reveals the correlation between values of n (from i=2 to i=10,000) and the corresponding Collatz sequence length. This code can be found starting from the first for-loop. It uses pipes to feed the data obtained from running the program into the wc - l command –it counts how many lines of code are in the output. This actually determines the length of any particular Collatz sequence for i because the collatz.c program writes each value of the sequence on a new line. The values of i and its lengths are redirected to a data file, where they can then be plotted with the specifications listed in the pseudocode listed above (for example, dots.)

2.2 Explanation of the Second Graph

The second plot shows the values of n with the corresponding maximum value of the Collatz sequence. The code for this can be found starting from the second for-loop in the pseudocode (from i=2 to i=10,000), and it also uses pipes. It takes the values obtained from running collatz.c (with i as the input), and sorts these numbers in reverse order using the command sort -nr. This is done by feeding the data into the command with the usage of pipes. The data is then again fed into another command (head -n 1) with a pipe to determine the first line of data. I used the command 'head' because once the numerical data is sorted in reverse order, the first line of the output would be the greatest value in the sequence. This would allow us to redirect both the values of i and its corresponding maximum number to a data file. Then this data can be plotted using Gnuplot with the ranges specified in the pseudocode and using dots.

2.3 Explanation of the Third Graph

As I already had the data file created from the first graph (that counts the length of the sequences), I simply reused it for plotting the third graph. I used awk to extract the second column of the data file and piped it into the command sort -n. The values in this data file are sorted numerically, and the frequency of each value is counted using the uniq command. Then these newly obtained values –that is, the frequency and the corresponding lengths– are appended to a new data file. This latter data file is the one that is used to make the frequency vs length plot. Once the data is entered into the second data file, the plot is graphed through Gnuplot with the specifications given in the pseudocode (draw using filled boxes and with the x and y columns of the data file switched.)

2.4 Explanation of the Fourth Graph

As mentioned earlier, I also created a fourth graph using much fewer data points. The code for this can be seen starting with the fourth for-loop. I ran the program 99 times randomly (from i=0 to i=99), and plotted the length of the sequence that is corresponded to each i's random run. The code for this is very similar to that of the first graph, however, because the collatz.c program uses the current time as the seed, the for-loop would only produce the same numbers each time if I didn't make additional changes. That is why I wrote the code sleep(1) –to ensure a new seed is generated for the program. I simply plotted the data with dots using Gnuplot.