

Krishna Sharma

CSE13s

Winter 2022 Long

01/09/2022

## Assignment 1 Design Document

### **Description of Program:**

This assignment asks you to create a `plot.sh` file that holds three for loops that each individually plot one of the figures from the assignment document. The `plot.sh` file is written in bash shell scripting, utilizing the given `collatz.c` file, for loops with bash scripting commands and `gnuplot` commands in order to produce the required plots.

### **Files to be included in directory “asgn1”:**

- `Collatz.c`
  - `Collatz.c` is the source file that contains the main and the C code that generates the collatz sequences.
- `Plot.sh`
  - Bash scripting file that contains the shell scripting commands to plot each of the figure graphs needed to complete the assignment.
  - Additionally it also contains the `gnuplot` commands needed to format each of the graphs.
- `Makefile`
  - File that formats program into clang-format and compiles it into program executable “collatz” with `make/make all` from `Makefile`.
- `README.md`

- Text file in Markdown format that describes how to build and run the program, how the program handles erroneous inputs, and any problems encountered while developing the program.
- DESIGN.pdf
  - Describes design for the program thoroughly with pseudocode and visualizations.

**Pseudocode / Structure:**

Forcibly clear the data file length.dat

For every sequence in range 1 to 10000

Print the sequence every time it loops and append it to the lengths.dat file

Take the sequence from the collatz file, generate the line count and append to the length.dat file

Exit loop

Forcibly clear the data file maximum.dat

For every sequence in range 1 to 10,000:

Print the sequence every time it loops and append it to the maximum.dat file

Take the sequence from the collatz file, sort numerically in reverse order and append to the maximum.dat file

Exit loop

Forcibly clear the data file histogram.dat

For every sequence in range 1 to 10,000:

Take the sequence from the collatz file, generate the line count, and append to the histogram.dat file

Exit loop

Concatenate the histogram.dat file, sort and count the numbers in this file and append to the histogram.dat file

Gnuplot END

Set terminal to pdf

Set output to length.pdf

Set title to Collatz Sequence Lengths

Set xlabel to x

Set y label to y

Set zeroaxis

Plot length.dat with points

END

Gnuplot END

Set terminal to pdf

Set output to maximum.pdf

Set title to Maximum Collatz Sequence Value

Set xlabel to n

Set y label to value

Set yrange to 10,000

Set zeroaxis

Plot maximum.dat with points

END

Gnuplot END

Set terminal to pdf

Set output to histogram.pdf

Set title to Collatz Sequence Length Histogram

Set xlabel to length

Set y label to frequency

Set xrange to 225

Set yrange to 200

Set zeroaxis

Plot histogram.dat with points

END

#### **Other Notes:**

##### **Collatz Sequence Lengths Figure 1:**

- The objective of this project is to create a graph that displays the collatz sequence lengths. A C program that produces collatz sequences is given.
- To start, all the lengths of a given collatz sequence should first be placed in a file. By getting the length of the collatz sequence and placing it into a length.dat file I will be able to easily obtain the line count and know how many lines were generated. The bash wc command will specify the line count of a file and this is needed to find the length of the collatz sequence, which is also what we need to create the y axis of the first graph. It is necessary to pipe the starting point, in this case \$n, into the wc to get the line count and then write it to /tmp/length.dat that way it is able to be read by gnuplot. At this time /tmp/length.dat only has the sequence length. In order to get an x coordinate it is necessary to echo x and place everything on the same line into a length.dat file for gnuplot to read. Echo x

naturally creates a new line; however, it is possible to not print the trailing new line character by adding a `-n`. All of this will be placed within a for loop that has a range of 1 to 10,000 as the x axis goes to 10,000 in the figure provided in the assignment 1 document.

- In the gnuplot commands it is important to change the output to `length.pdf`, the title to sequence lengths and the plot location to `length.dat` with points. For figure 1, it is important to take the existing program and format the output to what was needed; that way gnuplot is able to plot out the sequence lengths derived from the collatz sequence which is given. (Eugene Section 1/7/2022)

### **Maximum Collatz Sequence Value Figure 2:**

- For this figure, I will first need to write another gnuplot command which is similar to the one given for the collatz sequence length graph, that was modified to fit what was needed for the first figure. The gnuplot commands for this figure will also be slightly different. The output will be set to `maximum.pdf`, the title will be “Maximum Collatz Sequence Value”, the xlabel will be “n” and the ylabel will be “value”. For this graph I will also need to specify the y range by adding “set y range [0:100000]” which will extend my y range to what it needs to be. Gnuplot will also plot the location to `maximum.dat` with points.
- In order to create this figure, I will first need to get the maximum value of a sequence. By using a for loop to sort by the largest number and take the zeroth element we will be able to find the maximum value. If I use the sort command on the collatz file I should be able to see the maximum value of the sequence; however, there are two things to note. Firstly, it is necessary for me to pipe to less

that way I can parse through the file to see the values. Furthermore it is also good to note that when the `./collatz | sort | less` command is given, the values generated are shown lexicographically rather than numerically. I want to see the maximum value of the sequence, and thus I need the values to be sorted numerically. By adding `-nr` after `sort`, the numbers will sort in numerical order and they will be displayed in reverse, meaning that the maximum number will be at the top rather than at the bottom. (Eugene Section 1/7/2022)

### **Collatz Sequence Length Histogram Figure 3:**

- To start to create this figure I will first need to place all the sequence lengths into one file in order to see that there are clearly repeating numbers. In order to create a graph that displays a histogram of Collatz sequence lengths starting from  $n \in \{2, \dots, 10000\}$  I will need to `uniq` the repeated elements and then plot them. The `uniq` command in bash reports or filters out repeated lines in a file and it can also precede each output line with the count of the number of times the line occurred in the input if a `-c` is added afterwards. By doing this to the `length.dat` file I will be able to see how many times each element is repeated. The command `cat /tmp/length.dat | uniq -c | less` will be able to show me the exact amount of times an element is repeated in the collatz sequence lengths. (Eugene Section 1/7/2022)
- The `gnuplot` commands will also remain the same in structure for this figure; however, the contents will once again change, just like for figure 1 and 2. The `gnuplot` commands for this figure will have the output going to “`histogram.pdf`”, the title will be set to “Collatz Sequence Length Histogram”. The `xlabel` will be set to “`length`” and the `ylabel` will be set to “`frequency`”. `Gnuplot` will plot the

location to histogram.dat using 2:1 (meaning that the x should be column 2 and the y should be column 1, thus inverting the axis) with impulses rather than points.

**Cite:**

- I attended Eugene's section on 01/07/2022, which helped give me general guidance on how to approach this assignment. He went over the general concepts that are needed to successfully complete this assignment. His section helped me start this assignment and most of the concepts I used to complete it were pulled directly from his section. Most of my code for this assignment is a variation of what was shown during his section with small changes made to fit the specific constraints.