

Assignment 1
Gettting Acquainted with UNIX and C
DESIGN.pdf

Reuben T. Chavez

January 9, 2022

1 Description of Program

The following program plots three types of diagrams related to the Collatz sequence for numbers between two to ten-thousand. To give a brief description of the Collatz Sequence, it's a sequence of positive numbers, where the next number is determined by the previous number, and the sequence must always end with one. Where the next number is determined whether the previous number is a positive or negative. This function was used heavily within the program. The aforementioned program begins by first compiling the collatz.c program and removes necessary binary files, in order to rebuild the executable for collatz.c. After removing all files that are going to be used with in the program – to not create an output error when running again -, the program will begin in finding the data points for the Collatz Sequence Length. It will then store that data in .dat file so gnuplot can then plot it as a graph, that is stored as a pdf. It will do the same for the other two diagrams, the main differences being, but later explained, that they store and graph the data of the Maximum Number found in each Sequence and a Histogram that shows how frequent the length of a sequence is.

2 Files in asgn1

Brief Description of the following files are as followed:

- **collatz.c**: C program that simulates the Collatz Sequence formula and has the option to starts at designated number, or pseudo-random number
- **plot.sh**: Main program that calculate the data points for three different graphs with the collatz.c program and gnuplot to plot designated graphs and store in .pdf
- **Makefile**: File that stores commands to complies the collatz.c to executable collatz, format the program in clang format, and and removes any unnecessary binary files before the program, through the command make clean
- **README.md**: Marks Down File that list how the program can be run and built on the terminal. As well notes on how I handled error handling with the code
- **DESIGN.pdf**: A .pdf file that explains through pseudocode how the code is run and details info on all the other files within assignment one
- **WRITEUP.pdf**: A .pdf file that contains illustrations of the graphs produced through plot.sh and details on the trend and patterns of each graph

3 Pseudocode of plot.sh

The following lines are the steps plot.sh follows to plot all three diagrams

1. Access Makefile to rebuild executable for collatz sequence program
2. Remove files used in program
3. **Creating Collatz Sequence Length Graph Data**
For number from two to ten-thousand
 - (a) Call current number
 - (b) Append current number in file relating to the Collatz Sequence Length
 - (c) Run Collatz Sequence Starting at Current Number
 - (d) Find length of said sequences
 - (e) Append current sequence in in file relating to Collatz Sequence Length

End of iteration

4. **Plot Collatz Sequence Length**
 - (a) Call gnuplot
 - (b) Create and Name pdf
 - (c) Name Graph
 - (d) Set x axis and y axis range for Collatz Sequence Length
 - (e) Label x axis and y axis for Collatz Sequence Length
 - (f) Set increments for x axis and y axis for Collatz Sequence Length
 - (g) Plot the graph using the data points produced by the last iteration

5. **Creating Max Collatz Sequence Graph Data**
For number from two to ten-thousand
 - (a) Call current number
 - (b) Append current number in file relating to the Max Collatz Sequence
 - (c) Run Collatz Sequence Starting at Current Number
 - (d) Sort current sequence in numerical and reverse order
 - (e) Search for the number on the first line
 - (f) Append mentioned number in file relating to Max Collatz Sequence

End of iteration

6. **Plot Max Collatz Length**

- (a) Call gnuplot
- (b) Create and Name pdf
- (c) Name Graph
- (d) Set x axis and y axis range for Max Collatz Sequence
- (e) Label x axis and y axis for Max Collatz Sequence
- (f) Set increments for x axis and y axis for Max Collatz Sequence
- (g) Plot the graph using the data points produced by the last iteration

7. Creating Collatz Sequence Length Histogram Data

For number from two to ten-thousands

- (a) Run Collatz Sequence Starting at Current Number
- (b) Find length of said sequences
- (c) Append current sequence in in file relating to Collatz Sequence Length Histogram
End of iteration
- (d) Print Current File and pipe to next step
- (e) Sort File in reverse and numerical order, then pipe to next step
- (f) Count only the repeated sequences, then pipe to next step
- (g) Remove any unnecessary backslashes or empty space in file
- (h) Append file to self to store new data points

8. Plot Collatz Sequence Length Histogram

- (a) Call gnuplot
- (b) Create and Name pdf
- (c) Name Graph
- (d) Set x axis and y axis range for Collatz Sequence Length Histogram
- (e) Label x axis and y axis for Collatz Sequence Length Histogram
- (f) Set increments for x axis and y axis for Collatz Sequence Length Histogram
- (g) Plot the graph using the data points produced by the last iteration

Notes on Pseudocode and Structure

- The First Two lines are essential for the program as:
 - The Make file(line 1) make sure that collatz program is compiled and runs properly, while also removing unnecessary files

- The removed files(line 2) are there to make sure the program run smoothly each time it called, if wasn't there the code would be appending the same data to the same file making it the code to take longer to run
- The code itself use three iterations (3,5,7) each of which call the Collatz Sequence program to store and find data of the respected graph.
- Call current number (3.a., 5.a.) refers to printing the current number the for loop is iterating
- Append current number (3. b., 5. b.) in file relation ... means that number you just printed will be append in a .dat file with `;;` command
 - Said .dat file should be named something easy to remember as it will be used later to in gnuplot to plot the graph
 - Same thing applied with any line that starts with Append (3.e., 5.f., 7.c., 7.h)
- The lines with finding length (3.d., 7.b.) implement the `wc` command from UNIX. As the data to be store in a .dat retain how long the sequence for numbers being run in the collatz program
- Lines with sort in reverse and numerical order (5. d., 7. e) means that the current .dat, or the information to be appended to the .dat file is be reorganized with highest number at the top page to lowest number on the end of the page, which was implemented with UNIX sort command
- Lines with search for number on top (5.e.) means the implementation of UNIX's head command to search designated files first line, and used to find the max number after step (5.d.)
- Count only in the repeated sequences (7. f.), means the implementation of UNIX `uniq` command to count only the repeated sequences in the .dat file to find how many occurrences a certain sequence length was found after iterating the for loop
- The line to remove any unnecessary backlashes (7. g.) means the implementation of UNIX's `sed` command to format .dat file and remove any unnecessary empty space created by the `uniq -c-d` command (7.e.)
- With the plotting the data of the graph (4,6,8) the steps were ideally the same to initial `plot.sh` to plot the sine graph
 - Call plot (4.a, 6.a, 8.a) means to write the command `gnuplot;;END`, which signifies that gnuplot will follow the next commands until it sees the END command
 - Create and Name pdf means to (4.b,6. b,8. b) to call gnuplot's commands set terminal pdf and set output

- Name graph (4.c,6.c,8.c) means to name the graph through gnuplot's command set title
- Set x axis and y axis (4.d,6. d,8. d) means to set up the range of the x and y axis through gnuplot's command set xrange and set yrange
- Set increments for x axis and y axis (4.e,6. e,8. e) means to change the increments of the graph through gnuplot set xtics and set ytics
- Plot the graph (4. f., 6. f., 8. f.) mean to use the gnuplot's command plot along with the .dat file and other parameters to change how the graph stores the data – lines, dots, boxes, etc. The beginning and end of the for loops are written in italics to state that the list of orders between those lines are being done within the for loop, if they are outside the for loop, then it means there being done after the loop is finished

4 Credit

- I got the idea over my implantation from a response on ED in which someone was asking how the assignment one is supposed to work <https://edstem.org/us/courses/16730/discussion/983886>
- I also attended Eugene's section last Friday 11/7/22 where he gave me the early ideas on how to implement the plot.sh
- I used resources on discord under the asgn-1-faq-n-tips to format my DESIGN.pdf and WRITEUP.pdf
- This video to understand latex <https://youtu.be/Jp0lPj2-DQA>
- These video to understand gnuplot <https://youtu.be/9QUtcfyBFhE> <https://youtu.be/F_XcgIxdExE>