

Assignment 1 Design

Nhan Nguyen

January 2022

1 Program Description

The purpose of this program is to produce 3 different graphs based on the Collatz Sequence. The 3 graphs will be:

- A dotted graph that shows the lengths of each Collatz Sequence with $S_1 = \{2...10000\}$
- A dotted graph that shows the maximum value of each Collatz Sequence with $S_1 = \{2...10000\}$
- A histogram that shows the frequency of the lengths of each Collatz Sequence with $S_1 = \{2...10000\}$

This program is a bash script that utilizes a combination of C code with the provided code `collatz.c`, and the `gnuplot` tool in order to create the 3 different graphs.

1.1 List of files:

- **collatz.c:** The provided c program that produces a collatz sequence based on a number **n**
- **plot.sh:** The shell script program that produces the 3 graphs utilizing the provided `collatz.c` program as well as `gnuplot`
- **Makefile:** This file formats program into clang-format and also compiles it
- **README.md:** A text file in Markdown format that describes how to build the program, as well as describing the problems encountered while making the program
- **DESIGN.pdf:** This PDF file describes the idea behind the program with visualization and pseudocode
- **WRITEUP.pdf:** This PDF file contains all the graphs produced by the program as well as describing which command I use to generate the graphs

2 Ideas and Pseudocodes

2.1 Collatz Sequence Length dotted graph

- **Idea:** Since the provided C program will produce a collatz sequence based on number **n**. I utilized that behavior by printing the elements into a separate file then count the lines in said file, keep in mind that the number of lines will be stored in a parameter. After that, the parameter is printed to a file that is used as data to generate the graph through gnuplot. It's important to clear the file used to store the results from collatz.c because if not the data is going to overlap on each other which will mess with the results inserted to the file use for the gnuplot

-**Pseudocode:**

clear file **test.sh** to get rid of any data initially in there

for i in range from 2 to 10000:

 run collatz.c based on number **i**. Insert the result to a file called **test.sh**

 int count = the number of lines in **test.sh**

 print count into file **/tmp/collatz.dat**

clear file **test.sh**

Generate the dotted graph via gnuplot with the data in **/tmp/collatz.dat**

2.2 Collatz Sequence Maximum Value

- **Idea:** With the result provided by the program collatz.c. I inserted the results into a separate file. To find the maximum value of each collatz sequence, I sort the values within the file in descending as well as taking the value of the first line simultaneously through piping the commands. After that, the result is stored in a parameter and outputted to the file used for generating the dotted graph through gnuplot.

-**Pseudocode:**

clear file **test.sh** to get rid of any data initially in there

for i in range from 2 to 10000:

 run collatz.c based on number **i**. Insert the result to a file called **test.sh**

 int max = the result after sorting **test.sh** in descending order and taking the value of the first line

 print max into file **/tmp/collatzMX.dat**

clear file **test.sh**

Generate the dotted graph via gnuplot with the data in **/tmp/collatzMX.dat**

2.3 Collatz Sequence Length Frequency

- **Idea:** The idea of getting the length of each starting value from 2 to 10000 is the same as the idea presented to generate graph 2.1. The difference is that there needs to be an array that will serve to tally up the frequencies of each different lengths that will appear. This array will have an initial size of 225 with an initial value in each element of the array being 0. After the frequencies

of each lengths have been tallied up, we will output the values in each index (starting from 2) to a file in order to generate a histogram through gnuplot

-Pseudocode:

clear file **test.sh** to get rid of any data initially in there

initialized an array **freq**

for i in range from 1 to 225:

 freq[i] = 0

for i in range from 2 to 10000:

 run collatz.c based on number **i**. Insert the result to a file called **test.sh**

 int count = the number of lines in **test.sh**

 increase freq[count] by 1

 clear file **test.sh**

for i in range 2 to 225:

 print freq[i] into file /tmp/collatzFreq.dat

Generate the dotted graph via gnuplot with the data in /tmp/collatzFreq.dat