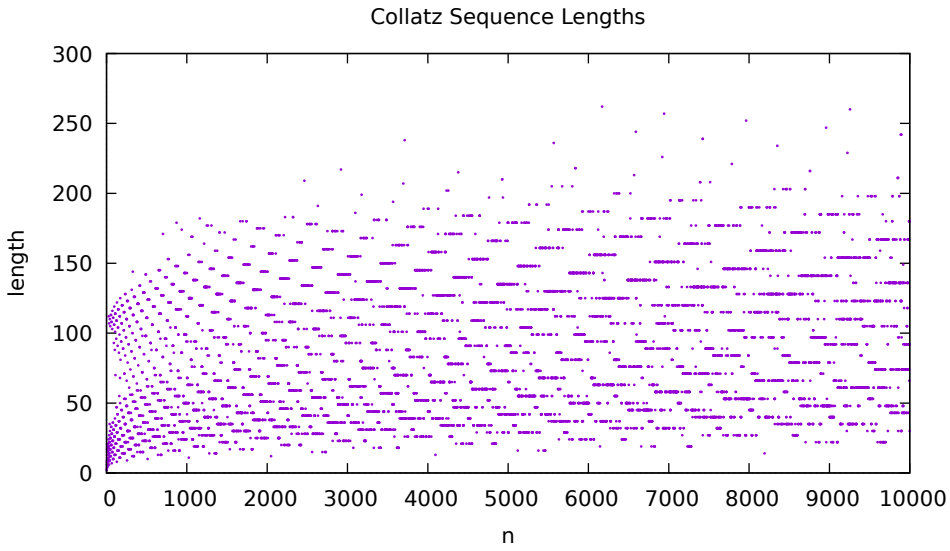
Assignment 1 - WRITEUP.pdf

Victor Nguyen

January 13, 2022

1 Plots and Analysis

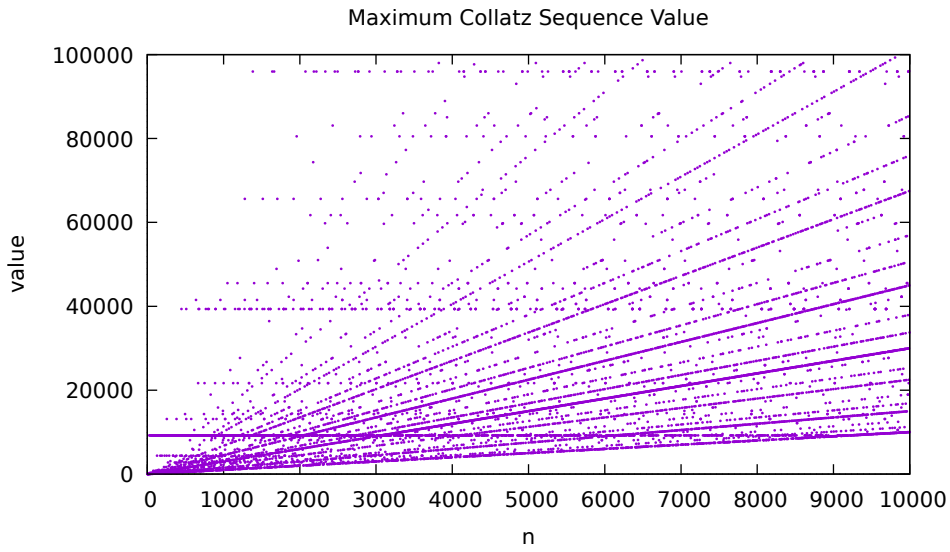
In this assignment, I was tasked to learn UNIX commands on the provided collatz.c file. The collatz.c file is the source code that generates a random collatz sequence (or a specific sequence if you know the proper syntax for it). Anyways, here are some of the graphs I produced with the provided source code.



This first graph depicts the relationship between a given collatz sequence n, and it's length. Commands I've used to produce this plot include for loops, executing collatz, echo, wc, and cat. Here are some reasons why:

- For loop - Needed to generate all collatz sequences from 2 to 10000.
- ./collatz - Needed to use the source code to generate the collatz sequences.
- echo - Needed to know what sequence number I was on and to put that on the graph.
- wc - Used to count the number of lines of a given sequence number.
- cat - Concatenate the length of the sequence number alongside the sequence number.

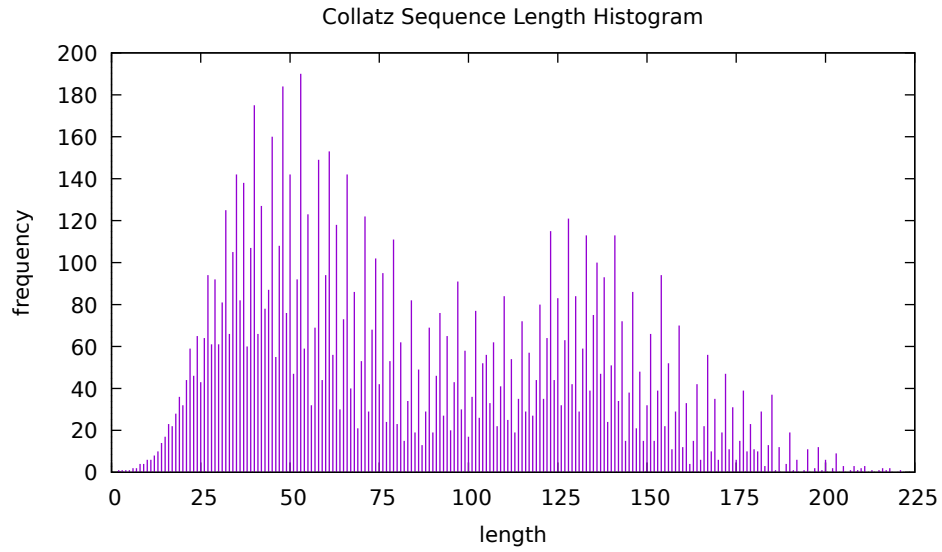
I've also used the gnuplot utility to take in a .dat file that I generated through this script, and use those data points to graph through gnuplot. Most of the gnuplot specifications were provided by the professor. Analyzing the graph, we can see that the growth of the sequence lengths gets smaller when we iterate to higher collatz sequences. It's growth is similar to the square root of x (if graphed).



This next graph depicts the relationship between a given collatz sequence n, and the highest value that the given sequence could rise to. Commands I've used to produce this graph include for loops, collatz, sort, head, echo, and cat. Some of these commands have the same usage as mentioned before, so to reduce redundancy I will only mentioned the new commands and their usage below.

- sort - Needed to sort the collatz sequence numerically so I can find the highest value easily.
- head - Used to grab the highest value in the given collatz sequence.

For the gnuplot, the only thing I changed was fixing the y axis to only look at numbers 0 to 100000. Analyzing the graph, for the most part, the growth of some numbers are linear. There are a few trends where some collatz sequences ending up with the same maximum value, producing a visible horizontal lines on the graph.



For the last graph, it depicts the relationship between the lengths of the collatz sequences (from 2 to 10000) and how many of those sequences match the same length of other sequences. Commands I've used to produce this graph include for loops, collatz, wc, sort, uniq, and awk.

- uniq - Used to count the occurrences of the same number.
- awk - Used to modify the .dat file and flip the x and y values.

For the gnuplot, I had to specify the x axis and also change the x axis tics to get the desired graph. Analyzing the graph, I noticed a high density of sequences with the lengths around 50. There also seems to be another peak in the 125 to 150 range. The frequency ranges tells me that most collatz sequences end within 200 "steps", and rarely last above that. This gives us slight evidence of collatz sequences always following that same 4 2 1 sequence.

2 Conclusion

From the data that we collected on the collatz sequences (ranging 2 to 10000). We can conclude that:

1. The higher the collatz sequence gets, the proportion between the length of the sequence "l" and that the value "v" increases, i.e. l:v over time becomes l:v + time.
2. The collatz sequence values grow in a somewhat, linear fashion.
3. Collatz sequences tend to only last around 200 steps before falling into the 4 2 1 sequence.