Guillermo Alfaro
CSE13S Winter 2022

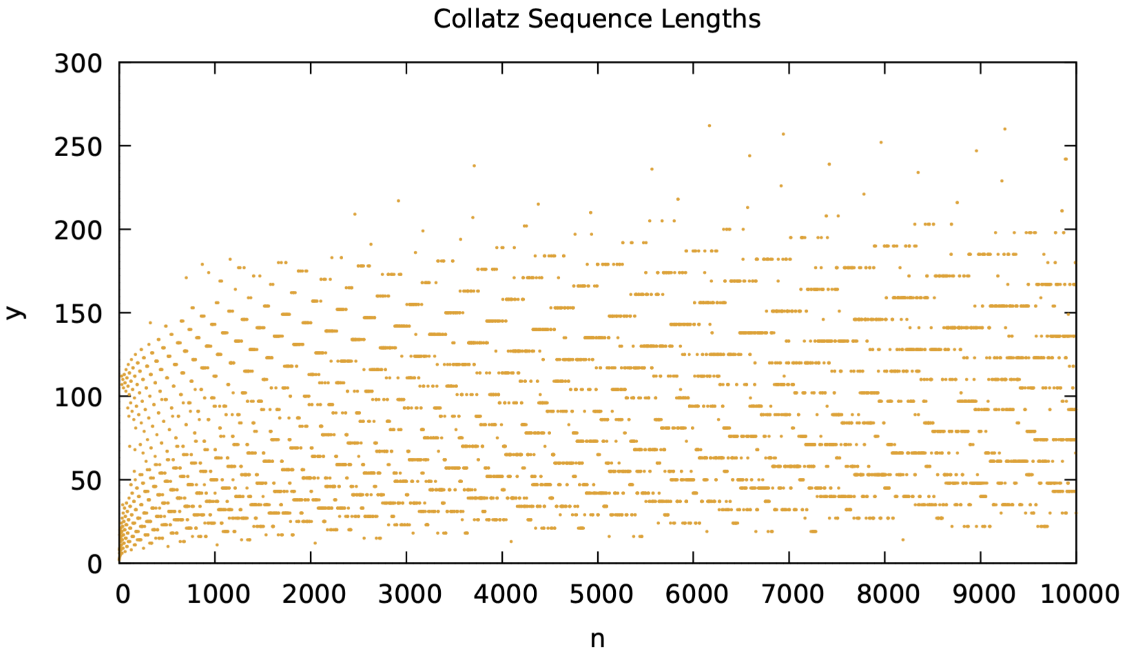
Assignment 1: Getting Acquainted with UNIX and C Writeup

Gathering Collatz Data:

To get the data I ran `$ ./collatz -n $i > collatz.dat` in a for loop starting at 2 till 10000. I then shifted the data around into other files before I cleared collatz.dat at the end of the cycle to make sure the last sequence didn't have data left behind.

Collatz Sequence Lengths:

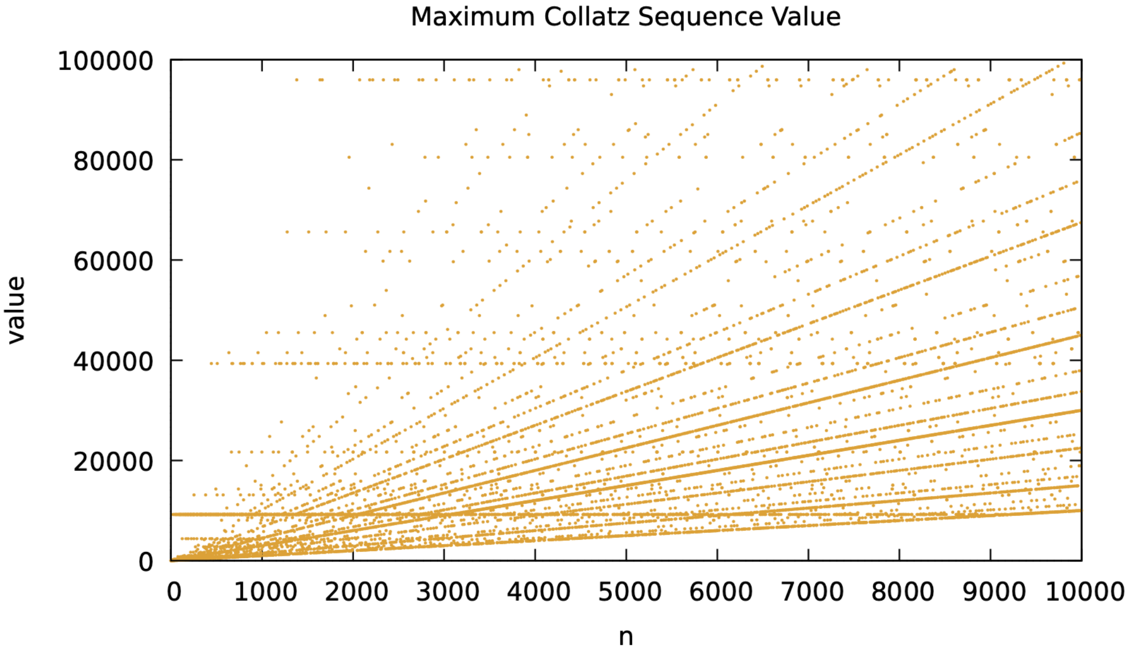
To get the length of each collatz sequence I used `$ wc -l < collatz.dat > datafile.dat` This command took collatz.dat, got the about of lines in it, then wrote it to datafile.dat `$ cat datafile.dat >> data.dat` then concatinated the amount of lines to the bottom of data.dat which was fed into gnuplot to produce this collatz sequence lengths.



After learning that UNIX has `$ wc` it was obvious to use it for lengths of a sequence, my original idea was to make a loop and increment a variable based on how many times it saw \n but that would have been unnecessary for my case.

Maximum Collatz Sequence Value:

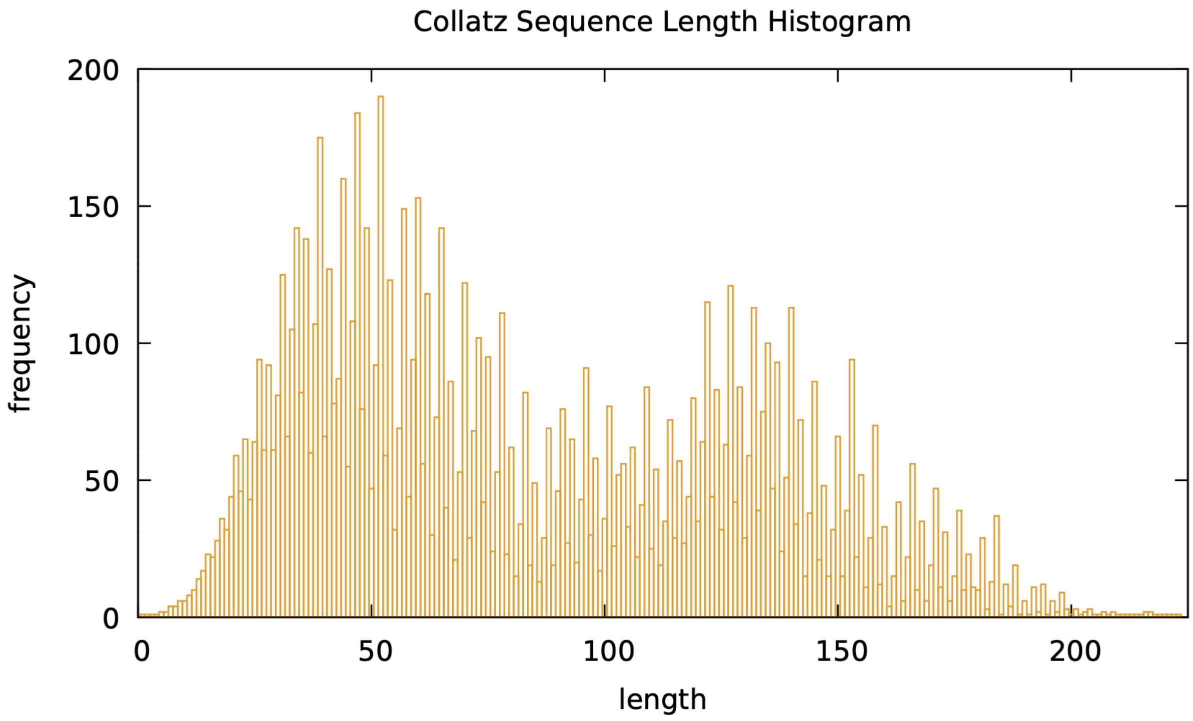
The maximum collatz sequence value was just another step to the collatz length. I used `$ sort -n -r -o sorted.dat collatz.dat` to sort all the numbers in collatz.dat in reverse and output them to sorted.dat. Then `$ head -n 1 sorted.dat >> moredata.dat` To get only the first line of the sorted which would be the maximum and copied that to the end of moredata.dat which was fed into gnuplot to produce this graph below.



Using `$ sort` in reverse mode was what i felt correct as it gives them in increasing order while only want the biggest. Then using `$ head -n 1` to get just the first line which would be the biggest and moving that to a file to get put into gnuplot seemed obvious.

Collatz Sequence Length Histogram:

For the histogram plot I needed to sort the original datafile.dat from the Collatz Sequence Lengths. Since I clear this file at the end of my for loop to be written to again. I concatinated it to a file simply called data.dat. Where I sorted that file with
`$ sort -g data.dat > placeholder.dat` then used
`$ uniq -c placeholder.dat >> histogram.dat` to count how many occurrences there were of each length where I copied that data to histogram.dat to be plugged into gnuplot.



Using `$ uniq -c` was always something I had planned on doing, but sorting the data by numbers before it was just one of those niche things I did to force the numbers I wanted.
`$ uniq` was only giving me the amount of times something repeated only right after another, so if a number didnt occur after itself it wouldnt be added to the uniq total. Sorting it to so every number is in its own group before hand was my solution.