

WRITEUP

Figure 2 Graph:

- For every graph I used a for loop that initializes the variable $n = 2$. I check if n is less than or equal to 10,000 since that is the range that is asked for. I increment n by 1 every iteration.
- I used the command `./collatz -n $n` which runs `collatz.c` with the starting number being whatever n is equal to on that iteration.
- For this graph I pipe the collatz sequence so that its output can be used as input for the following command.
- I pipe the collatz sequence into `wc -l` which only gives me the amount of lines or length of the sequence.
- This data gets written into a temporary file named `collatz2.dat` with `>>` so that the file isn't reset every time data is stored.
- This data gives me the lengths of each collatz sequence from `[2:10000]` and is plotted with dots.

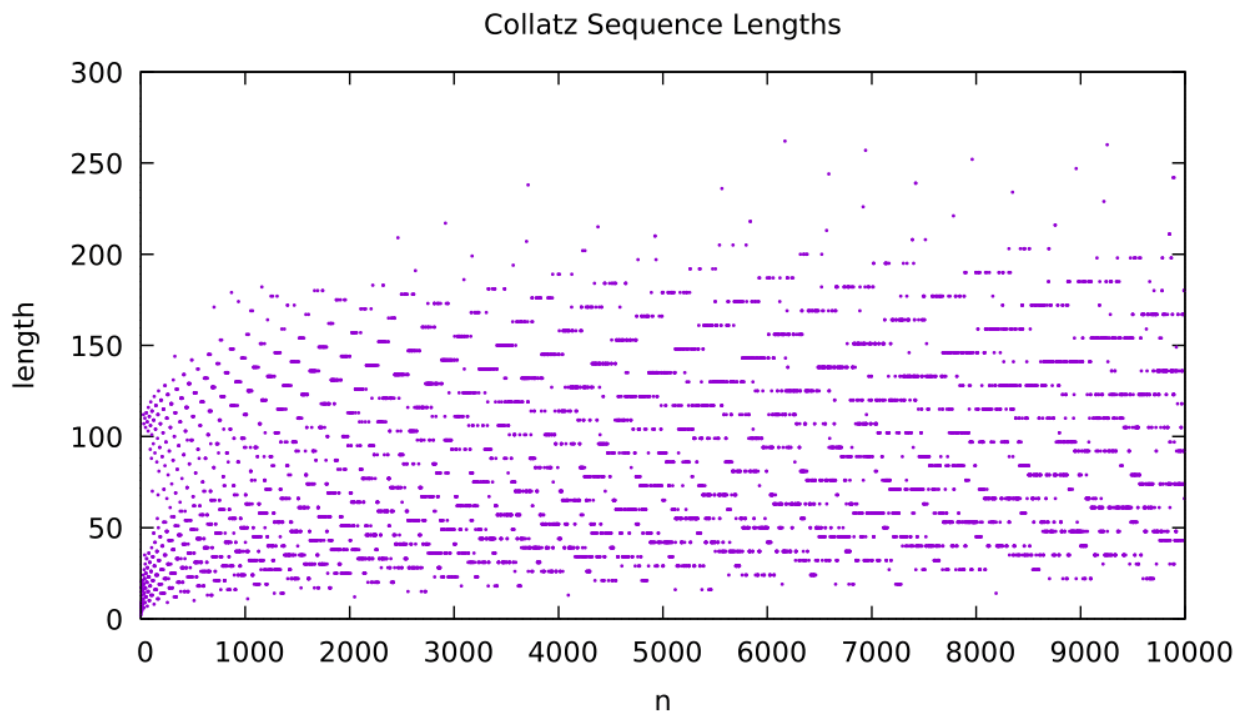


Figure 3 Graph:

- I used a for loop that initializes the variable $n = 2$. I check if n is less than or equal to 10,000 since that is the range that is asked for. I increment n by 1 every iteration.
- I used the command `./collatz -n $n` which runs `collatz.c` with the starting number being whatever n is equal to on that iteration.
- For this graph, I pipe the collatz sequence so that its output can be used as input for the following command.
- I pipe the sequence into the command `sort -n -r`. `Sort -n` numerically sorts the sequence from lowest to highest. `Sort -r` reverses the sorted values so that the largest value is on the first line.
- I then pipe the sorted numbers into the command `head -n 1` which only gives me the first line of the input which is now the largest value in the collatz sequence.
- The largest number is transferred to a temporary file named `collatz3.dat` with `>>` so that the file isn't reset each time.
- This gives me the largest value of each collatz sequence from $n = [2, 10000]$. The data is plotted with dots.

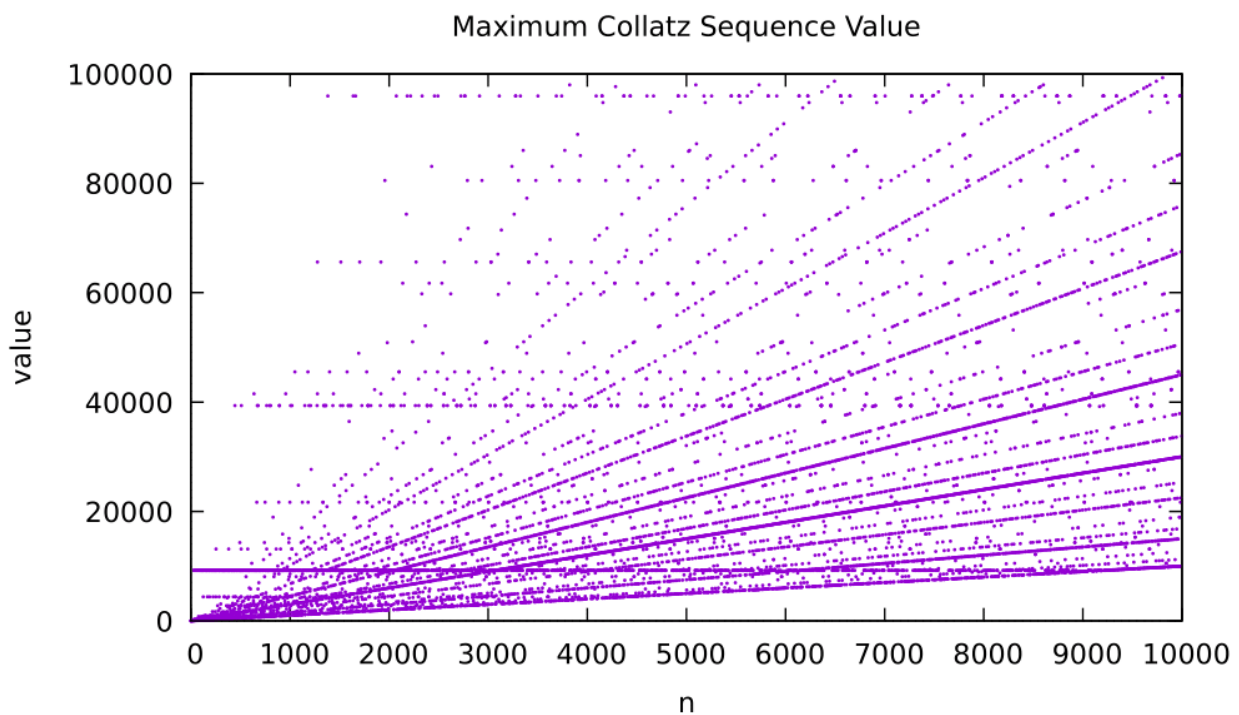
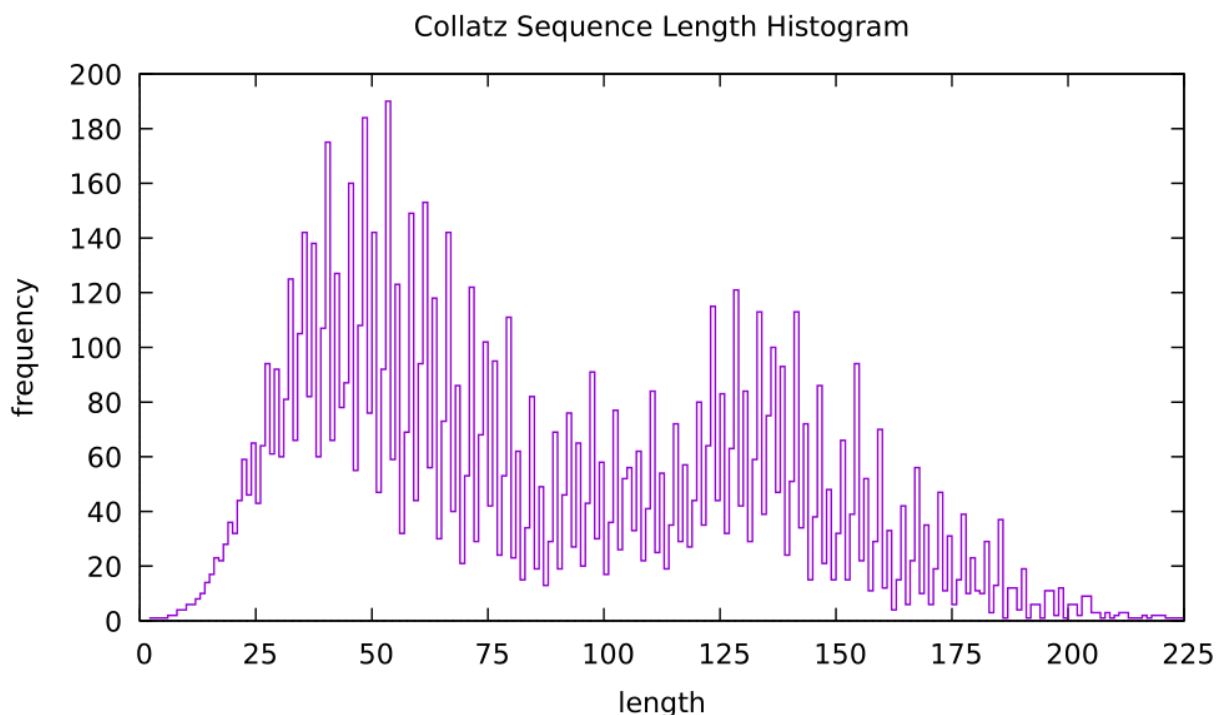


Figure 4 Graph:

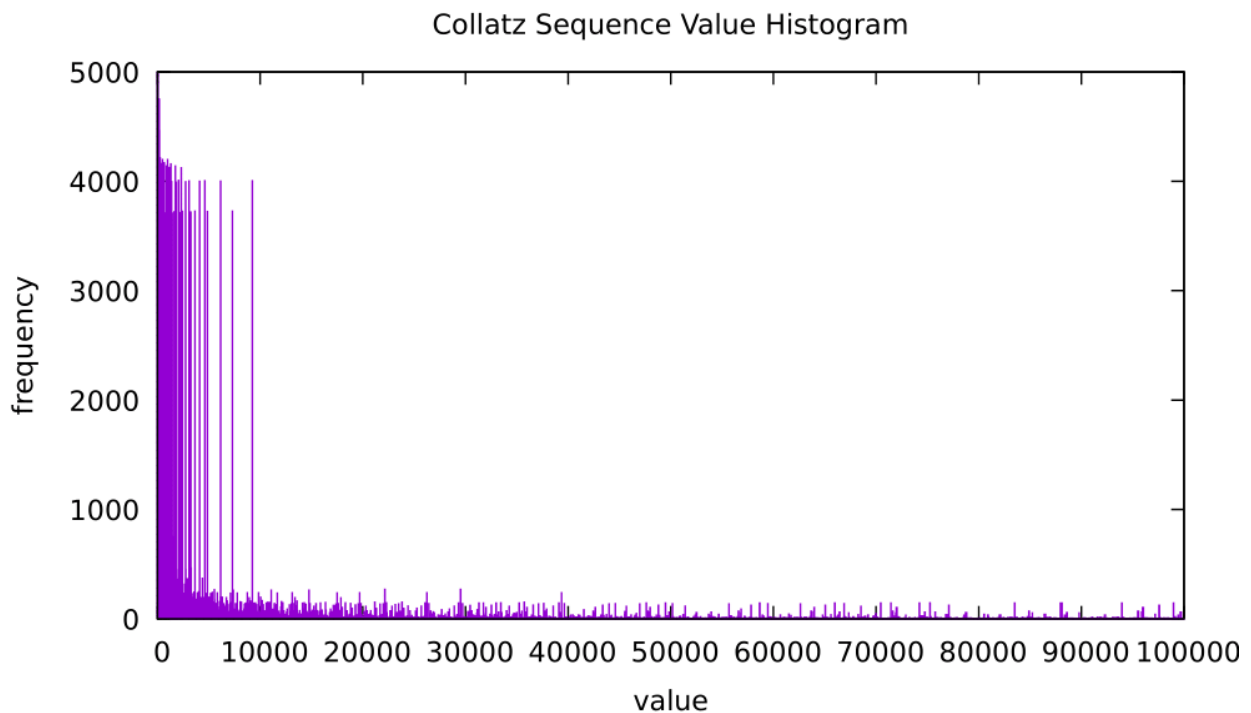
- For this graph I used the same temporary collatz2.dat file from the figure 2 graph since they use the same data.
- I numerically sort the collatz2.dat file from lowest to highest with `sort -n`. I write this data into a temporary file named collatz_p.dat with `>` so the file is reset every time plot.sh is run. I numerically sort the data since the next command needs sorted data to properly work.
- I use the command `uniq -c` on the collatz_p.dat file from before to get the amount of times each collatz sequence length occurs. This data is written into a temporary file named collatz4.dat with `>` so that the file resets every time plot.sh is run.
- The collatz4.dat file now has the data to plot the histogram but the frequency is on the x side of the data and the length is on the y side of that data.
- When plotting this data i write `# plot "/tmp/collatz4.dat" using 2:1 with steps`. Using 2:1 makes the x side of the data be used for the y-axis and the y side of the data be used for the x-axis. I plot with steps to get the same look of the figure 4 example.



Extra Graph:

- I used a for loop that initializes the variable `n = 2`. I check if `n` is less than or equal to 10,000 since that is the range that is asked for. I increment `n` by 1 every iteration.
- I used the command `./collatz -n $n` which runs collatz.c with the starting number being whatever `n` is equal to on that iteration.
- I write all the values of the collatz sequence into a temporary file named collatz_e.dat with `>>` so that the data isn't erased on every iteration.

- I numerically sort the collatz_e.dat file from lowest to highest with sort -n. I write this data into a temporary file named collatz_pe.dat with > so the file is reset every time plot.sh is run. I numerically sort the data since the next command needs sorted data to properly work.
- I used the command uniq -c on the collatz_p.dat file from before to get the amount of times each value in every collatz sequence from $n = [2:10000]$ occurs. This data is written into a temporary file named collatz5.dat with > so that the file resets every time plot.sh is run.
- The collatz5.dat file now has the data to plot the histogram but the frequency is on the x side of the data and the value is on the y side of that data.
- When plotting this data i write # plot "/tmp/collatz5.dat" using 2:1 with steps. Using 2:1 makes the x side of the data be used for the y-axis and the y side of the data be used for the x-axis. I plot with steps to get a histogram graph.



End of Plotting:

- I removed the files collatz2.dat, collatz3.dat, and collatz_e.dat with the command rm after all the data is plotted. This is because I used the >> keyword which doesn't erase the data in the file that is being written in. This means the data must be removed manually with the command rm at the end of the plot.sh file.