Assignment 1 - WRITEUP.pdf

Aniket Pratap

January 10, 2022

Abstract

This WRITEUP.pdf will include the design and execution process of assignment 1: collatz sequences. Before I start begin this "tail" I would like to explain the collatz sequence. The sequence consists of the equation:

$$S_{k+1} = \begin{cases} \{1 + S_k & \text{if } S_k \text{ is odd} \\ \frac{1}{2} S_k & \text{if } S_k \text{ is even}\} \end{cases} \quad (1)$$

The goal of this lab was to graph 3 things:

- Use a scatter plot to display the lengths of the sequence from n = 2-10,000
- Use a scatter plot to display the max sequence value
- Use a histogram to plot the colaltz sequenes and their frequency

1 Lengths of Sequence

The first problem I faced was understanding the assignment. I believed that the data had to be random every time so I struggled with the initial for loop. Adding on, I also didn't know why my collatz sequence didn't change when I ran it through the loop. After consulting Eugene, he told me that the initial collatz number was being randomly set by the time, and that a pseudo random number would form every second. The probelm with my loop was that it ran faster than a second–meaning that I got that same starting value over again. After rereading the assignment, I discovered that n was the starting value. This allowed me to change my whole approach. I first began the bash script with the flag:(number-sign) !/bin/bash. This flag is used to tell the computer that this is a bash script. My next step was creating a for loop: for ((i=2; i <10001; i++)) so that 10,000 is included. I then ran ./collatz along with -n and the iterator variable so that I print the data on a new line without the trailing.The wc -l flag printed the new line count–so by piping ./collatz's output, I was able to obtain the lengths. I then appended that data, using », to a file called /tmp/length.dat–recomended by Eugene as it deletes data when the program wasn't in use. I had a little trouble aquiring the xcoordinates because when I did echo with the for loop number, the xcoordinate and ycoordinate printed out as one. My solution to this was quite lucky and I simply asked myself, "what if I put a space after the incrementor and tada! It worked. My coordinates were now seperated and graphable using gnu. The gnu code was slightly altered to accomodate for the labels but it stayed relativly the same. The only big change I made was using dots to plot the data. The final Collatz Sequence Length Graph:

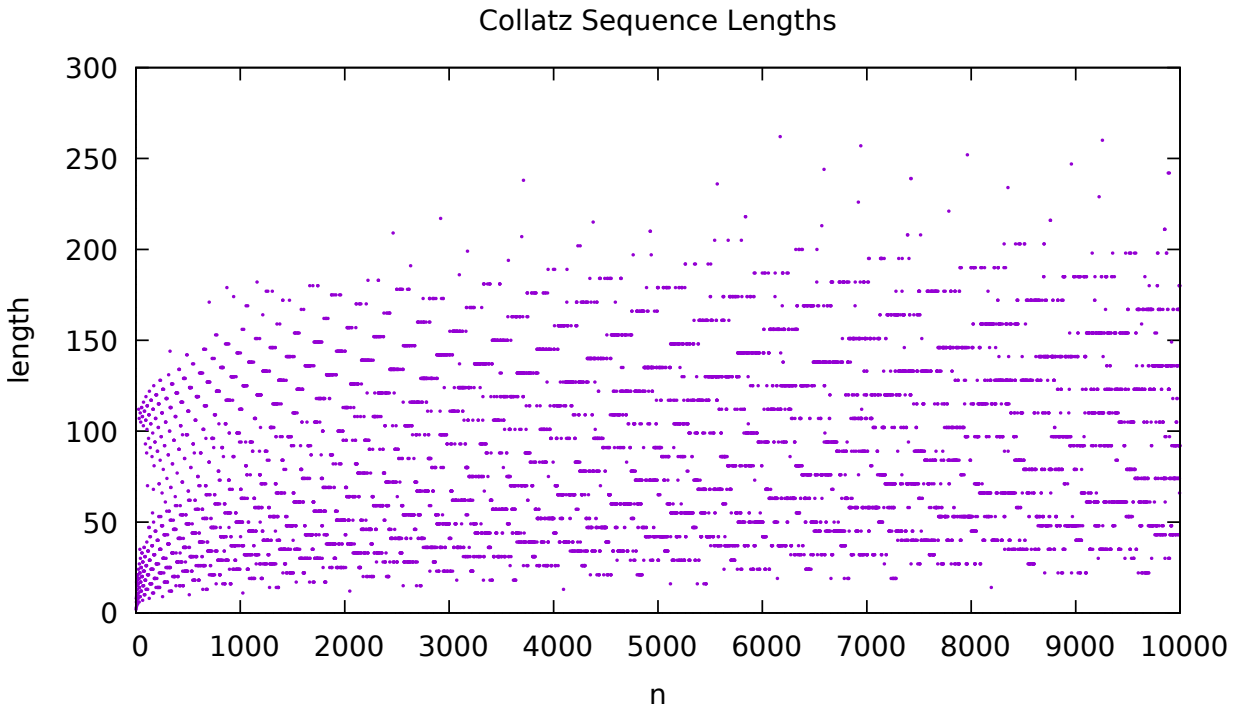

Figure 1: Length of Collatz Sequences from 2-10,000

2 Maximum Sequence Value

As I finished the first program successfully, the maximum sequence in each list of numbers seemed more doable. I used my same tactic in order to obtain the x-coordinate but this program had to run so that out of the list of collatz numbers, the maximum is recorded in a data file. Using | sort -n, I was able to sort the list of numbers numerically instead of lexicographically. The larest number would no be at the end of the list. In order to obtain it, I used tail -n 1 to specifiy that I only want the last line. After being piped with the previous statement and appended to /tmp/max.dat I was able to store and plot the max number in each sequence:

3 Collatz Sequence Histogram

I had the most trouble with this section because my data wasn't outputing to a file properly. I sarted with storing all the data into a file called /tmp/hist.dat. I did this inside the for lop so I could collect the length of all the sequences first, otherwise uniq -c would count each run seperatly. To make debugging easier, I sorted the data using sort -n then piped it with uniq -c and tee. I used tee, because gnu plot nor awk would be able to detect 2 different columns–they only detected it as one. Not only that, the x-coordinates would dissapepar for some reason. Based on this reasoning, if I se tee, which read from standard input and to standard output, I was able to paste the data into the same file. Freaky? I don't know but I didn't come up to this conclusion on the spot, but after a 3 hour period scouring through the man pages and lecture slides. seen as that worked, I placed the statement outside of the for loop so that it could call uniq

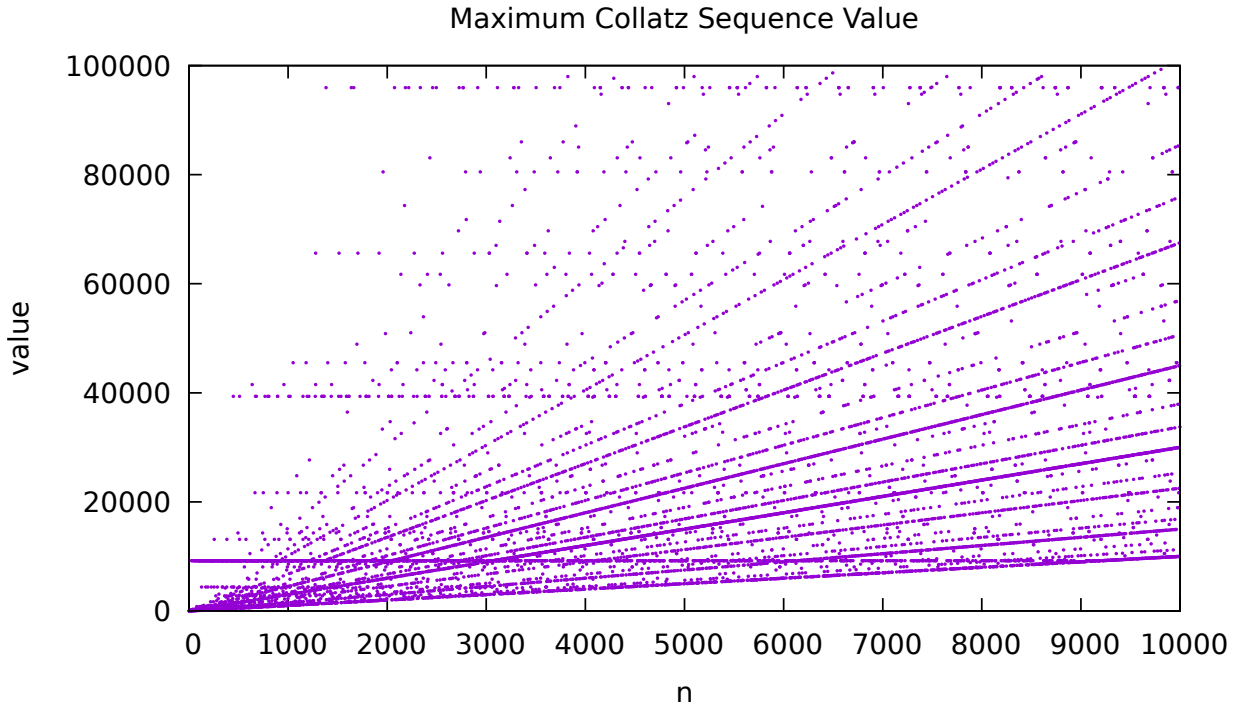

Figure 2: Max Values from 2-10,000

-c on the total data. After that, I noticed that I needed to reverse the coordinates in order to obtain the frequency so I used "using 2:1" in order to achieve this–courtesy of Eugene. This however, did not work with hisograms so I found a different ploting method, boxes, and set the width to a small number.Seen as this replicated the examples, I was satisfied. I also added clean and make collatz so that the compiled files would be removed at each run–along with compiling collatz. Not only that, but I removed the daat graphs, also courtesy of Eugene, so that everything would be fluid and automated without any manuel deletions. Histogram with frequencies:

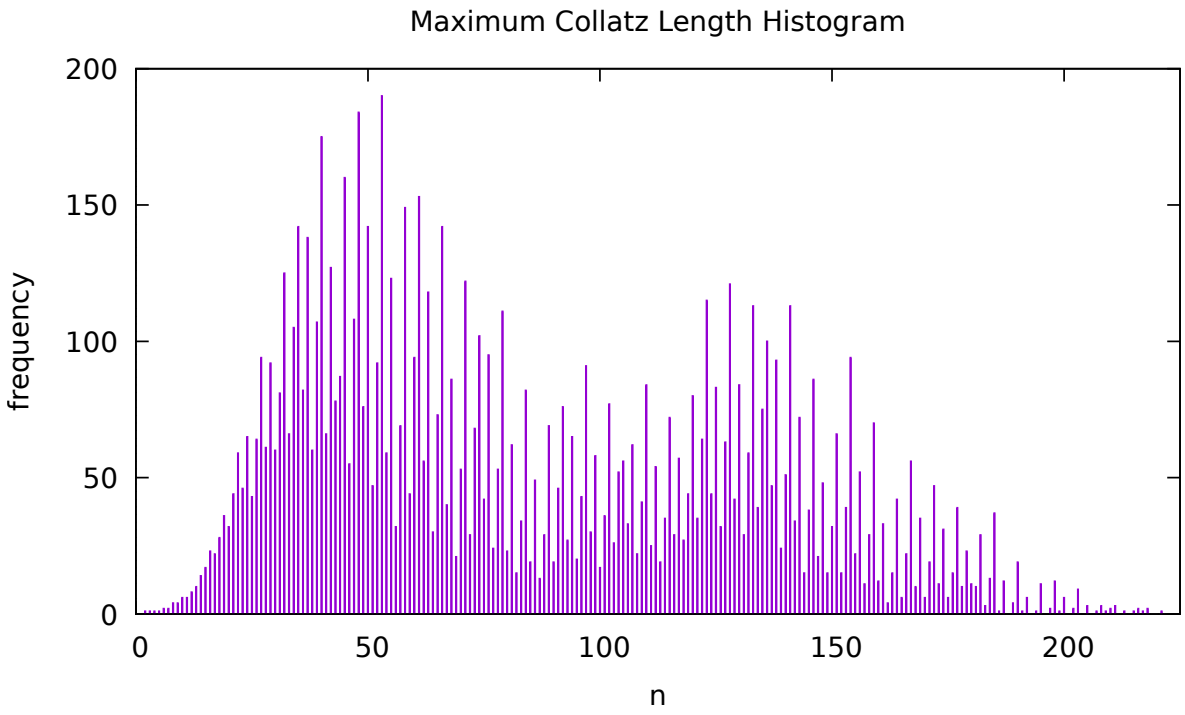

Figure 3: Histogram of Collatz Sequences from 2-10,000