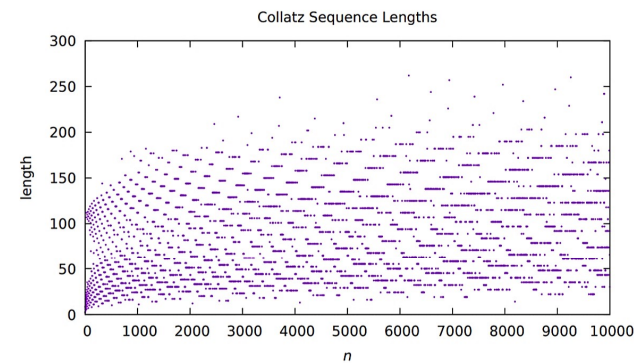
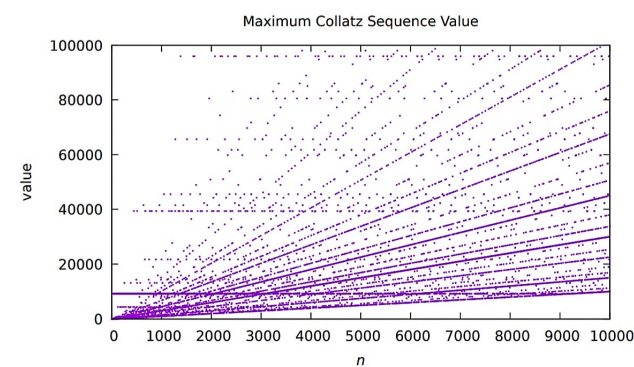


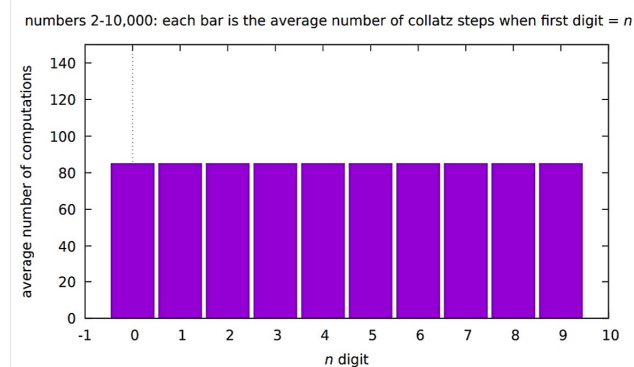
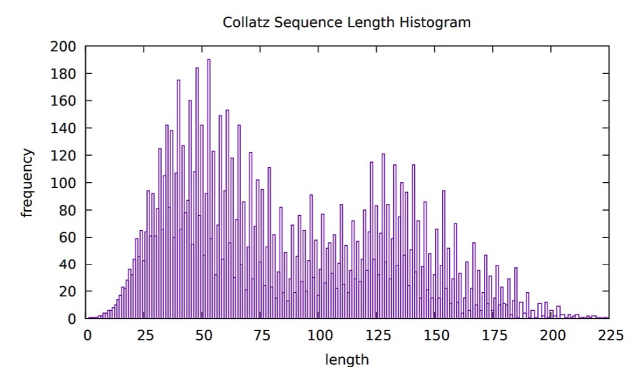
PDF Write-up



Plotting computation steps against number was probably the easiest of the plots. I learned both pipes and wc from the assignment instructions. I intuited I would need a for loop to use the element variable as an argument to collatz. Then when I was looking example.dat I realize line number would be equivalent to computation step number. So then I piped it all together and it worked.



This plot was only slightly harder. I figured "sort" from assignment one would be helpful in sorting the data numerically. I also figured there was something like head but for the end of the data. (I suppose I could have used head and sort -nr but I just didn't) I looped through the collatz sequences sorted the longest one to the bottom, piped that data to tail, and appended it to my output data file. It was very intuitive.



This creative one was pretty difficult too, it describes an interesting mathematical phenomenon of equality even though it outputted some really plain looking results. In this one I used a for loops to iterate over the digits and a nested while loop to iterate over every length there could be from 2-10,000. In the nested if statement I used parameter expansion to output a string one character long one characters away from the end. I did this because I knew there would be a newline and I wanted the first digit of the number.

The frequency vs length was by far the hardest graph for me. It took me a day to figure out exactly what my pseudo code was going to even be. Soon I realized I could use a memory variable and it all fell into place. I used a while loop and read to iterate over the actual data and I used tail again to chop of the newline at the top of my collatz_2.dat to clean it up for my computation. I used double brackets if statement and -eq to test my variables for numerical equality. And lastly echo with -n to stop trailing newlines.