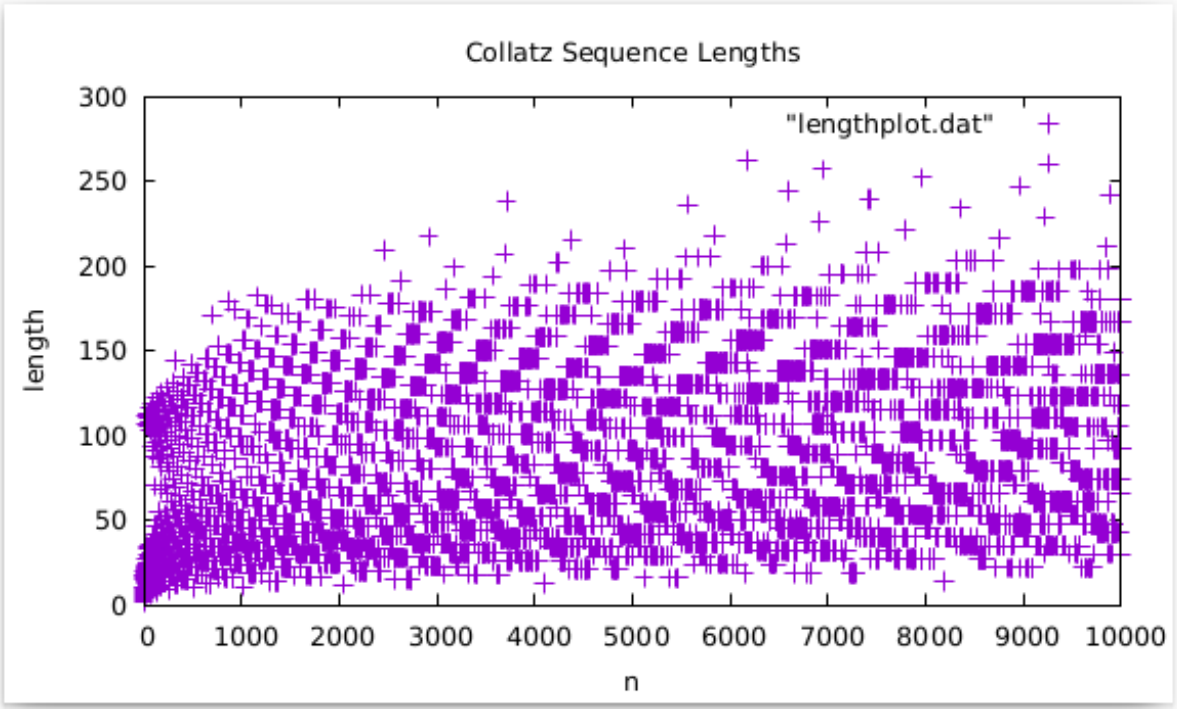
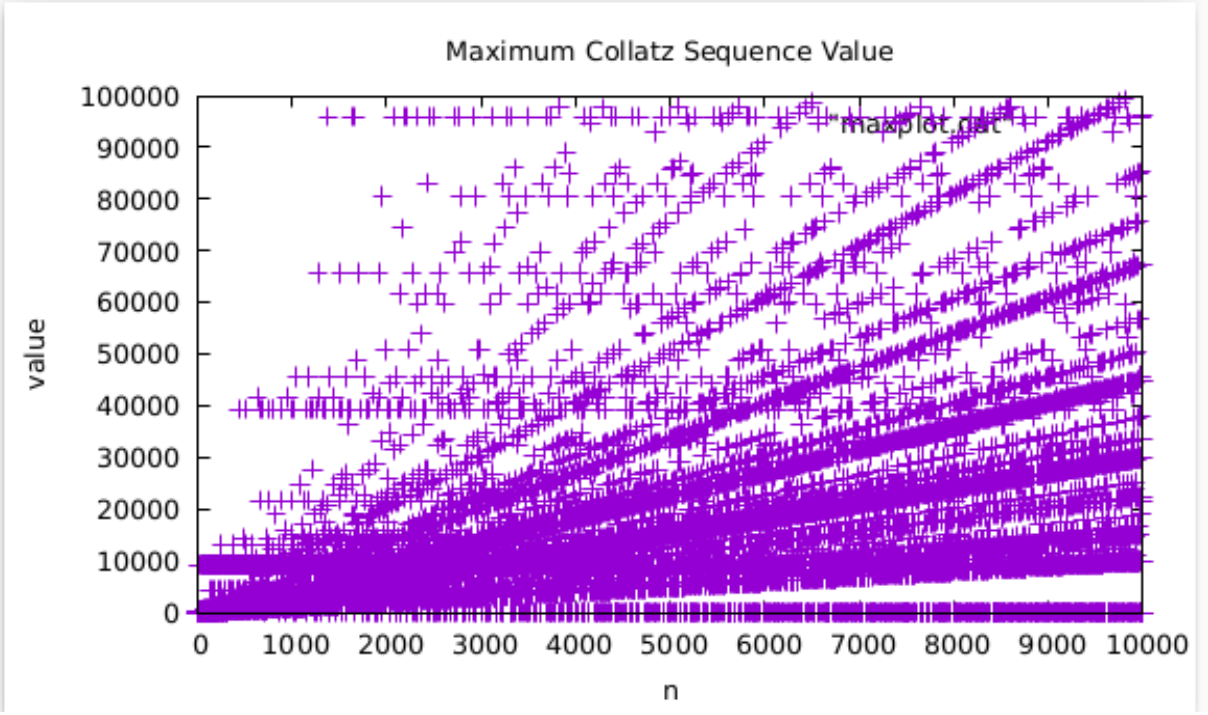
CSE13s WRITEUP:
Liam Murray
Lijamurr



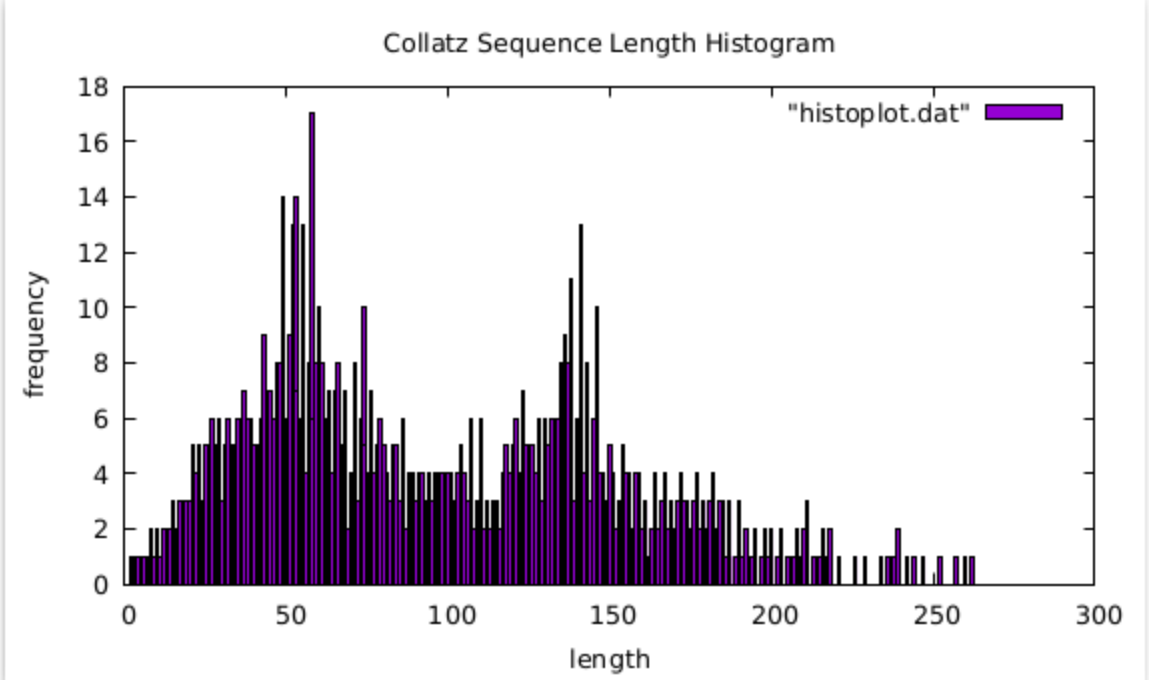
This is the Collatz Sequence Lengths graph. This graph represents the total length of a given sequence (y) and the number on which that sequence started (x).

Commands Used:
Head: I selected to use the “head” function, as it allowed me to pipe the input of a file directly into the next command, allowing me to easily move data without having to rerun the “Collatz” executable.
Wc: I selected to use this command as it could determine the length of an inputted file, which would in turn produce the length of the Collatz sequence.
Echo: Echo allowed me to format the data in (x, y) coordinate pairs to be inserted into the file used by gnuplot.
For loop: The for loop allowed me to iterate over the desired collatz sequence values, inputting the value into a coordinate function.



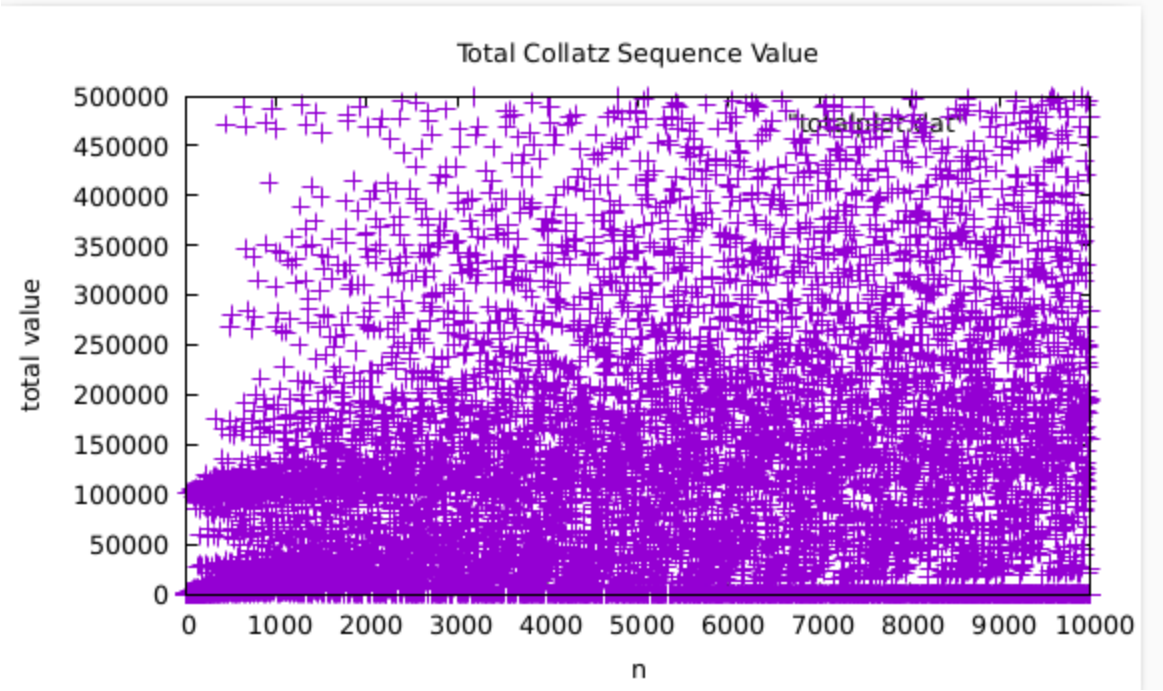
This is the Collatz Sequence Maximum Value graph. This graph represents the maximum value of a given sequence (y) and the number from which that sequence started (x).

Commands Used:
Sort: I selected to use the “sort” function, as it allowed me to organize the data numerically, which allowed me to skim the first number in the file to find the largest number.
Head: I used this function as it allowed me to read the top number of a data file, which was helpful when dealing with numerically sorted data files.
If, then: I selected to use this command as it could determine if the “maximum value” of a sequence was over a certain threshold. Oftentimes extreme outliers could skew data, and removing them allowed for a more meaningful and understandable graph.
Echo: Echo allowed me to format the data in (x, y) coordinate pairs to be inserted into the file used by gnuplot.
For loop: The for loop allowed me to iterate over the desired collatz sequence values, inputting the value into a coordinate function.



This is a histogram of how often certain lengths of collatz sequences appeared in a data set. It shows the length value being recorded (x) and the frequency with which that data point occurred (y).

Commands Used:
Head: This command allowed me to pipe the collatz sequence from a file into another command.
WC: This command allowed me to determine the length of a file, and thus the length of the collatz sequence.
Echo: I chose this command as it allowed me to format the data into appropriate (x, y) coordinate pairs in a very simple way.
For loop: The for loop allowed me to iterate over the desired collatz sequence values, inputting the value into a coordinate function.
Uniq: This function allowed me to discover how often data appeared in a file. It listed the number of times a data value appeared, followed by the data value itself. I was able to feed this data into gnuplot to produce the histogram.
Awk: I used this function to swap the data output of Uniq from (y, x) to (x, y). This allowed me to feed the appropriate data to gnuplot and not have a reversed histogram.
Sort: This function allowed me to make sure that the data was in numerical order, so that the x-axis would be more readable.



This is a graph of the total added value of every number in a collatz sequence (y), and its starting value (x).

Commands Used:
Declare: This function allowed me to store data in a variable as explicitly numerical. This means that when I added a value to it, it would be treated as a math expression rather than a concatenation.
Head: This function allowed me to pipe the collatz sequence data from a file into another command.
For loop: I used this because it provided a convenient way to iterate through the required “n” (collatz sequence starting point) values.
If, then: This function allowed me to catch data that would skew my plot (such as values that totalled to be over 4,500,000) and remove it, leaving a more readable and useful graph.
Echo: I used this command to format the data into (x, y) pairs to be sent to gnuplot.