

Assignment 1 Writeup

Nhan Nguyen

January 2022

1 Introduction

For this writeup, I will describe the commands I used in my bash script **plot.sh** in order to generate the 3 graphs, the graphs generated are based on the Collatz Sequence. The 3 required generated graphs are:

- A dotted graph that shows the lengths of each Collatz Sequence with $S_1 = \{2...10000\}$
- A dotted graph that shows the maximum value of each Collatz Sequence with $S_1 = \{2...10000\}$
- A histogram that shows the frequency of the lengths of each Collatz Sequence with $S_1 = \{2...10000\}$

Note that the program is completely written in a bash script and utilizes the provided C program **collatz.c**

2 Code and Graph

2.1 Collatz Sequence Length

- **Bash:** There are a few components that made up the code for this graph. The required components are: bash's version of a for loop since we need to generate the length of each collatz sequence with numbers from 2 to 10000, **wc -lines** to count the number of lines in the file. Since the collatz.c program prints out the collatz sequence based on the starting number n, using **wc -lines** basically means to get the length of that collatz sequence, ">" **sign** as a way to clear out the result file outputted from **collatz.c** as well as the file use the generate the graph using gnuplot.

- **Gnuplot:** The type of graph used to generate is the dotted type, with xrange, yrange set to automatic

- **Bash code:**

```
iter=2
> /tmp/collatz.dat
```

```

for i in {2..10000}
do

    ./collatz -n $iter > test.sh
    cnt='wc --lines < test.sh'
    printf "%i\n" $cnt >> /tmp/collatz.dat
    ((freq[$cnt]++))
    > test.sh
    ((iter++))

done

```

Note: This code has a portion of the code used for generating the histogram but the functionality is mostly dedicated for generating the Collatz Sequence Length dotted graph

- Gnuplot code:

```

gnuplot <<END
    set terminal pdf
    set output "collatz.pdf"
    set title "Collatz Sequence Length"
    set xlabel "n"
    set ylabel "length"
    set zeroaxis
    plot "/tmp/collatz.dat" with dots
END

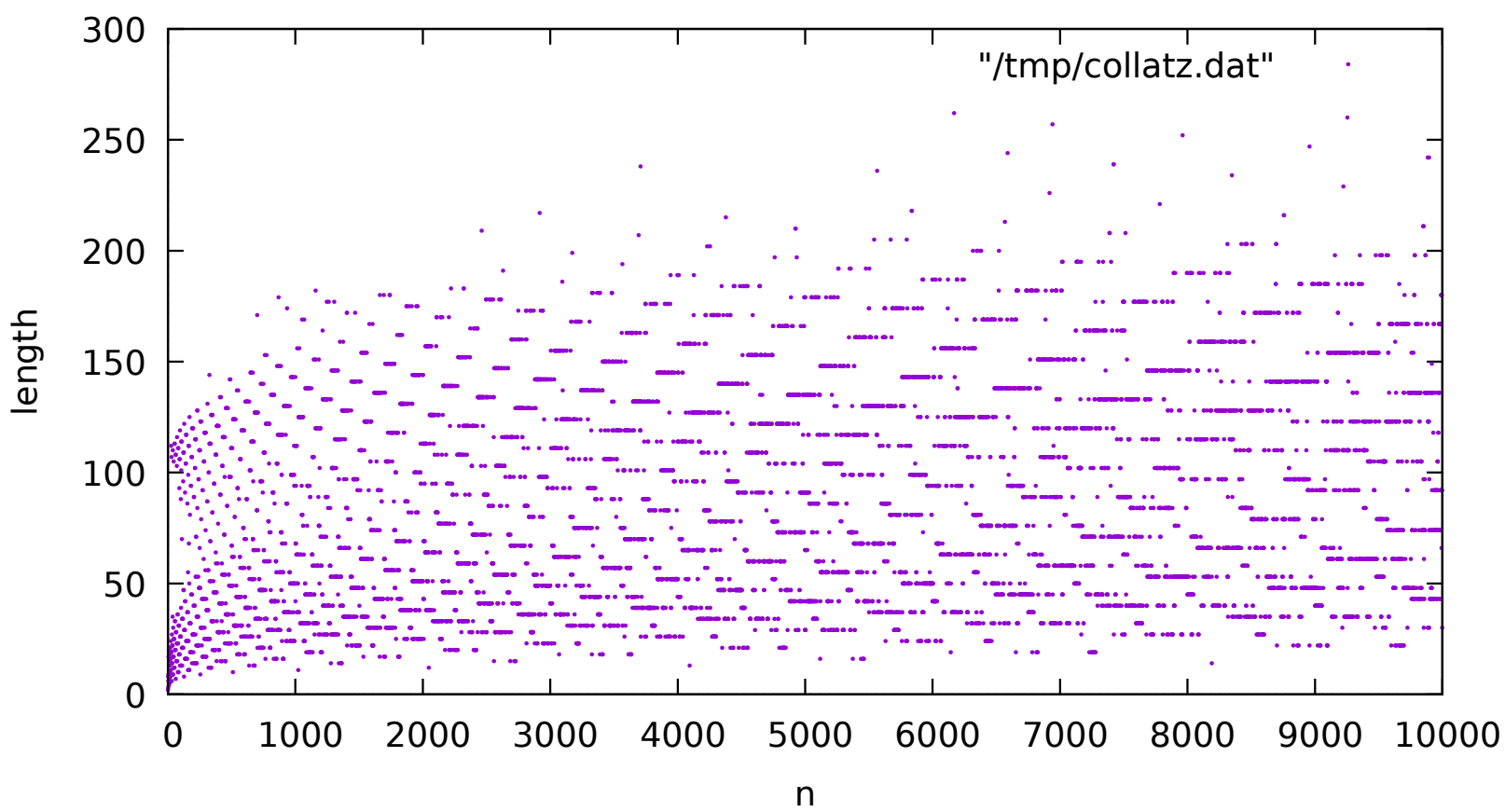
```

- **Difficulties:**

Testing whether or not the graph is correct or not is the most difficult part when making this algorithm because I need to create an empty test file to see whether or not the bash code generates the correct data for gnuplot to produce the graph. If there is leftover data remaining in the file the gnuplot uses, it will result in an incorrect graph. Therefore after many tests, I realized that I need to clear out all the files before inserting data from collatz.c to ensure the gnuplot generates the graph correctly.

- **Plot generated by Gnuplot and Bash:**

Collatz Sequence Length



2.2 Collatz Sequence Maximum Value

- **Bash:** The components in this program are mostly the same as the components used in generating the graph in **2.1**, Since this graph focus on the maximum value of each collatz sequence, the difference is that instead of using **wc -lines**, I will be using 2 important components: bash's sorting algorithm in descending order piped with the **head -n 1**. Since when sorted in descending order, the first value is always going to be the largest value, paired up with the command **head** which gets the value of the first line, the result will be the maximum value of the collatz sequence.

- **Gnuplot:** The type of graph used is the dotted graph which has the yrange specify to a range from 0 to 100000. The reason I manually set the yrange is because there's an error where the gnuplot plot sets the yrange too high automatically.

- **Bash Code:**

```
iter=2
> /tmp/collatzMX.dat
for i in {2..10000}
do

    ./collatz -n $iter > test.sh
    mx='sort -rn test.sh | head -n 1'
    printf "%i\n" $mx >> /tmp/collatzMX.dat
    >test.sh
    ((iter++))

done
```

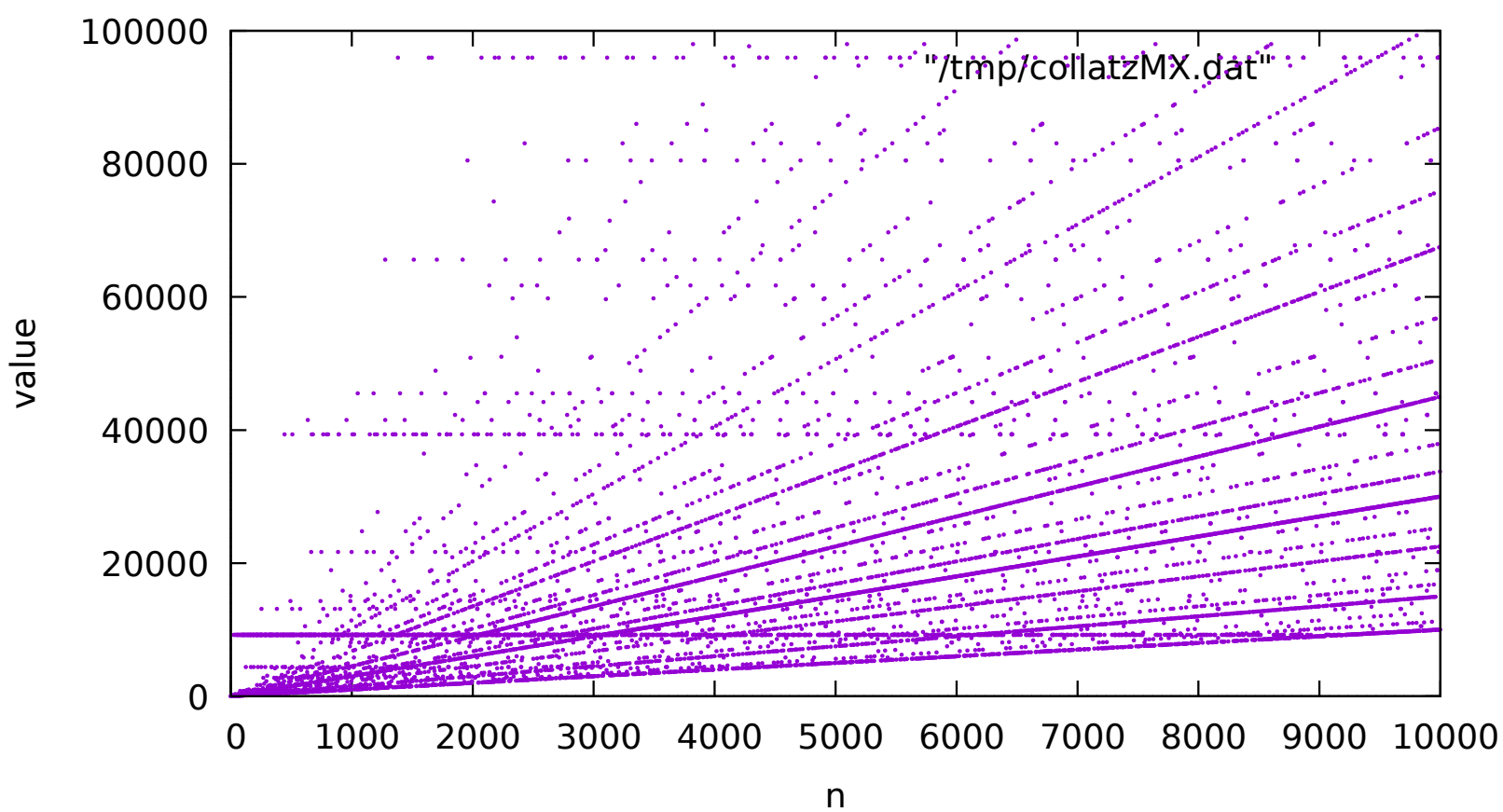
- **Gnuplot Code:**

```
gnuplot <<END
    set terminal pdf
    set output "collatzMX.pdf"
    set title "Collatz Sequence Maximum Value"
    set xlabel "n"
    set ylabel "value"
    set yrange [0: 100000]
    set zeroaxis
    plot "/tmp/collatzMX.dat" with dots
END
```

- **Difficulty:** I believe the main difficulty I encountered when making this algorithm is figuring out which method I'm going to choose. At first I was going with the method of using an additional for loop in the original for loop to go through each value in the file and update the **mx** parameter accordingly. The method proves to be too hard to implement, and that compelled me to go with the sorting method I had now.

- **Plot generated by Gnuplot and Bash:**

Collatz Sequence Maximum Value



2.3 Collatz Sequence Length Histogram

- **Bash:** The components used to generate this graph is the same as the components used in generating graph **2.1**. However, I used an additional component called the **freq** array with an initial size of 225 and a starting value of 0 for each element. The reason I used the **freq** array is because I need something called a counting array, which is an array that keeps track of the number of values that appear while running the algorithm to generate the collatz sequence length data for graph **2.1**. In other words, this **freq** array will give me the frequency of each kind of length from 2 to 225 while generating the data for graph **2.1**

- **Gnuplot:** The type of graph used is called impulse (histogram). Additionally the xrange is customized to a range from 0 to 225, and the yrange is customized to a range from 0 to 200. I manually set the xrange and yrange just to ensure the graph comes out correctly

- **Bash code:**

```
freq=()
for i in {0..225}
do
    freq[$i]=0
done

iter=2
> /tmp/collatz.dat
for i in {2..10000}
do

    ./collatz -n $iter > test.sh
    cnt='wc --lines < test.sh'
    printf "%i\n" $cnt >> /tmp/collatz.dat
    ((freq[$cnt]++))
    > test.sh
    ((iter++))

done

> /tmp/collatzFreq.dat
for i in {1..225}
do

    echo ${freq[$i]} >> /tmp/collatzFreq.dat

done
```

- **Note:** The code used here is also a part of the code used in **2.1**, this is the full code used to make the data used to generate the histogram.

- **Gnuplot code:**

```
gnuplot << END
    set terminal pdf
    set output "collatzHis.pdf"
    set title "Collatz Sequence Length Histogram"
    set xlabel "length"
    set ylabel "frequency"
    set xrange [0:225]
    set yrange [0:200]
    plot "/tmp/collatzFreq.dat" with impulses
END
```

- **Difficulties:** The main difficulty is figuring out how to generate the array with a certain size and with a specific starting value for each element. Additionally, figuring out which type of graph in gnuplot is used to produce the histogram poses a big problem cause I've tried more than 5 different types of histograms until I settled with impulses

- **Plot generated by Gnuplot and Bash:**

Collatz Sequence Length Histogram

