CS4115 Week04 Lab Exercise

Lab Objective: The objective of this week's lab is to continue with last week's tasks. By the end of this lab you should have

❶ implemented as programs the chunks of code from Q. 2.7 of Chapter 2 of *Weiss*;

❷ recorded in a spreadsheet, for each program, its average running time; and

❸ plotted these running times in `gnuplot`

In Detail

❷ Time your programs' running time using the `time` command as described last time. You should pick values of n spaced well apart and run the program about 5 times for each value of n to get a reliable average running time from them. It is best to space the values of n out according to a log scale so appropriate values of n (when permitting) might be $n = 10^2, \ldots, 10^6$. Use a spreadsheet row for each value of n and record the 5 running times for each value of n in adjacant cells on the row. Then compute the average of these alongside. So the rightmost column should give a fairly reliable running time for each of the values of n.

In the spirit of the Roman *Divide et Impera* maxim if you like you can share this task with *one* friend by each of you doing some of the data collection, but you must make it clear who you collaborated with.

❸ You should now be able to plot these in `gnuplot` by saving the values of n and the average running times to a file, say, `q271.dat`. The format of the file should be of the form

```
10 0.34
100 1.7
   :
```

The `gnuplot` command to plot this is then

```
plot [10:10000] 'q271.dat'
```

Play around with plotting different curves along with the recorded data so see if you can find the constants associated with a file of data. So, for example, you might try the following plotting commands to estimate the polynomial associated with the data in the file '`q271.dat`'.

```
plot [10:10000] 'q271.dat', x**2
plot [10:10000] 'q271.dat', 10*x**2
plot [10:10000] 'q271.dat', 7*x**2
              :
```

A handy trick to know in `gnuplot` is that Ctrl-p ("control p") repeats the last command and you can edit it.

We will return to this one more time and consider how we could bring a bit more science to how we estimate our measured running times as a function of n.