Integer Factorization

Alejandro Medina Diaz

19/09

1 Objectives

This particular project began with understanding the actual purpose of factorizing a number, and the use of the subprogramme factorize was the kay.

It was firstly to complete a sheet-to-code page, in which the professor claimed us what he had to do on the code, and part by part, with some intuition and interpretation, was done.

After implementing the algorithm, the next step was to integrate the solution with the provided project setup in Eclipse. This involved configuring the build path correctly and setting the necessary files and folder structures.

I also had to changes between concrete numbers, then files to try on the code, even if I had the accuracy that the code was correct, because the machine gave us a mark.

Even though my computers are mid-class, what matters here is the quality code, in which, though, I tried to be as efficent as possible

2 Experimental Setup

I used the options -n for running through specific numbers, -t for files, which made it more interesting since I could try on many different numbers simultaneously, and finally, when the stats.txt was finally developed, was had to navigate through RStudio, which, luckily is not a new gadget for me.

All of this happens on ECLIPSE IDE setup, concrettely on JAVA SE 17. The machine used for testing has a quad-core processor, 16GB RAM, and runs a 64-bit Windows 10 operating system. This environment provides enough resources to handle the input files efficiently and measure performance across different numerical values.

(see Table 1).

Table 1: Computational environment considered.

CPU	Intel Core i7-9700K @ 3.60GHz, 8 cores
RAM	16GB DDR4
OS	Windows 10 Pro, 64-bit, version 21H1
Java	Java SE 17

3 Empirical Results

A summary of the experimental results is provided in Table 2 in the Appendix, along with the statistical fitting of the data to different growth models.

Describe the results, in particular Figure 1.

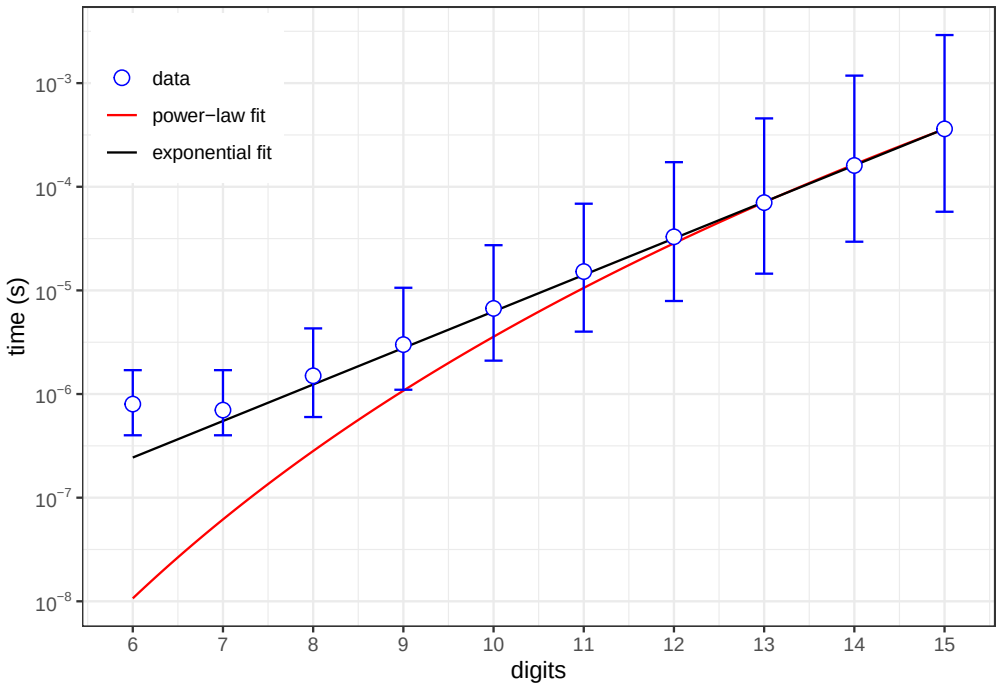

Figure 1: Time required for factorizing integers of increasing number of digits

4 Discussion

Provide your interpretation of the results: discuss whether the results match the theoretical predictions, whether some algorithm is better in practice than others, etc.

A Appendix

A.1 Data Summary

Table 2: Summary of the experimental results. Q1, Q2 and Q3 represent the 1st, 2nd (i.e., median) and 3rd quartile respectively. All times are in seconds.

#digits	time (Q1)	time (Q2)	time (Q3)
6	4.00e-07	8.00e-07	1.70e-06
7	4.00e-07	7.00e-07	1.70e-06
8	6.00e-07	1.50e-06	4.30e-06
9	1.10e-06	3.00e-06	1.06e-05

#digits	time (Q1)	time (Q2)	time (Q3)
10	2.10e-06	6.70e-06	2.73e-05
11	4.00e-06	1.52e-05	6.86e-05
12	7.90e-06	3.29e-05	1.73e-04
13	1.45e-05	7.03e-05	4.57e-04
14	2.95e-05	1.60e-04	1.18e-03
15	5.74e-05	3.62e-04	2.91e-03

A.2 Model Fitting

A.2.1 Power-Law Fit

```
## Power law fit
## --------------------

##
## Formula: time ~ a * digits^b
##
## Parameters:
##    Estimate Std. Error t value Pr(>|t|)
## a 1.490e-17  7.980e-18   1.867   0.0989 .
## b 1.138e+01  1.990e-01  57.173 9.72e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.06e-06 on 8 degrees of freedom
##
## Number of iterations to convergence: 23
## Achieved convergence tolerance: 2.853e-06
```

A.2.2 Exponential Fit

```
## Exponential fit
## ------------------

##
## Formula: time ~ a * b^digits
##
## Parameters:
##    Estimate Std. Error t value Pr(>|t|)
## a 1.877e-09  9.346e-11   20.08 3.94e-08 ***
## b 2.251e+00  7.590e-03  296.52  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.537e-07 on 8 degrees of freedom
##
## Number of iterations to convergence: 3
## Achieved convergence tolerance: 5.21e-08
```