

Computer Science 2300: Lab 1

Due: February 3, 2010

This lab is intended to ensure that you are comfortable with the tools you'll be needing throughout the semester – the appropriate GNU compiler, and a Unix-like shell environment. It will also allow you to explore the practical difference between an exponential algorithm and a linear one in terms of how fast they are in computing Fibonacci numbers.

1 Lab Problem

Implement two separate programs that generate the n th Fibonacci number. The programs should both take n as command line input. The first one should implement the recursive algorithm described in class, and the second one should implement the iterative algorithm, except that it should compute the n th Fibonacci number *one million* (10^6) *times* (although you only need to print it out once). You'll see the reason for this once you execute the program – it is so fast that it's hard to get a meaningful difference in times for different n without repeating the process many times!

Once the two programs are implemented, use the `time -p` command (typically in a bash shell generated by cygwin, Mac OS X or some other Unix-like distribution) to measure the time it takes for each of these versions to calculate the n th Fibonacci number (for the iterative implementation, divide the output of the `time` command by 10^6) for the following values of n :

1, 5, 10, 15, 20, 25, 30, 35, 40, 41, 42, 43, 44, 45, 46, 47, 48

For example, if the name of the executable is `fib`, then the following command would give you the running time for calculating the 10th number in the Fibonacci series:

```
$ time -p ./fib 10
```

Finally, create a table of all your results and use your favorite plotting software to create two scatter plots of the time taken to compute the Fibonacci numbers using the two different algorithms. Show your results and your program to your TA to receive credit.