Lab 1: Fibonacci Series

Due Wed 9/8/21;

Please attend to only your lab section!

Part 1:

Implement two separate programs that generate the n-th Fibonacci number. The programs

should both take n as command line input.

The first algorithm one should implement the recursive algorithm (fib1 on Page 3 of the

textbook), and the second one should implement the iterative algorithm (fib2 on Page 4).

You should measure the time it takes for each of these versions (use for example,

the time it module in python) to calculate the n-th Fibonacci number for the following values of n:

1, 5, 10, 15, 20, 25, 30, 35, 40, 41, 42, 43.

Finally, create a table of all your results and plot the time taken to compute the Fibonacci numbers using fib1 and fib2, with n on the x-axis and time

on the y-axis.

Part 2:

Now verify that fib2 is indeed quadratic time. Run fib2 and record the time on the following

values of n: 2^{10}, 2^{12}, 2^{14}, 2^{16}, 2^{18}, 2^{19}

If it takes more than 20 mins on your machine to compute 2^{19}, then drop the last value above.

For this part you can appreciate the fact that Python has native arbitrary precision integer

arithmetic. Otherwise, the values of F(n) would easily exceed the 64 bit word size and cause

overflow.

Plot the time of fib2 versus n.

Grading

Your code must be implemented in Python. To get full credit, you must show your code, output,

and plots to the TA and/or the undergrad mentors. Make sure that your grade is recorded in the

official sheet. The lab will be graded according to the rubric outlined on the syllabus. Do not

submit your code via email or any other means to the TA/mentors. It will be ignored.