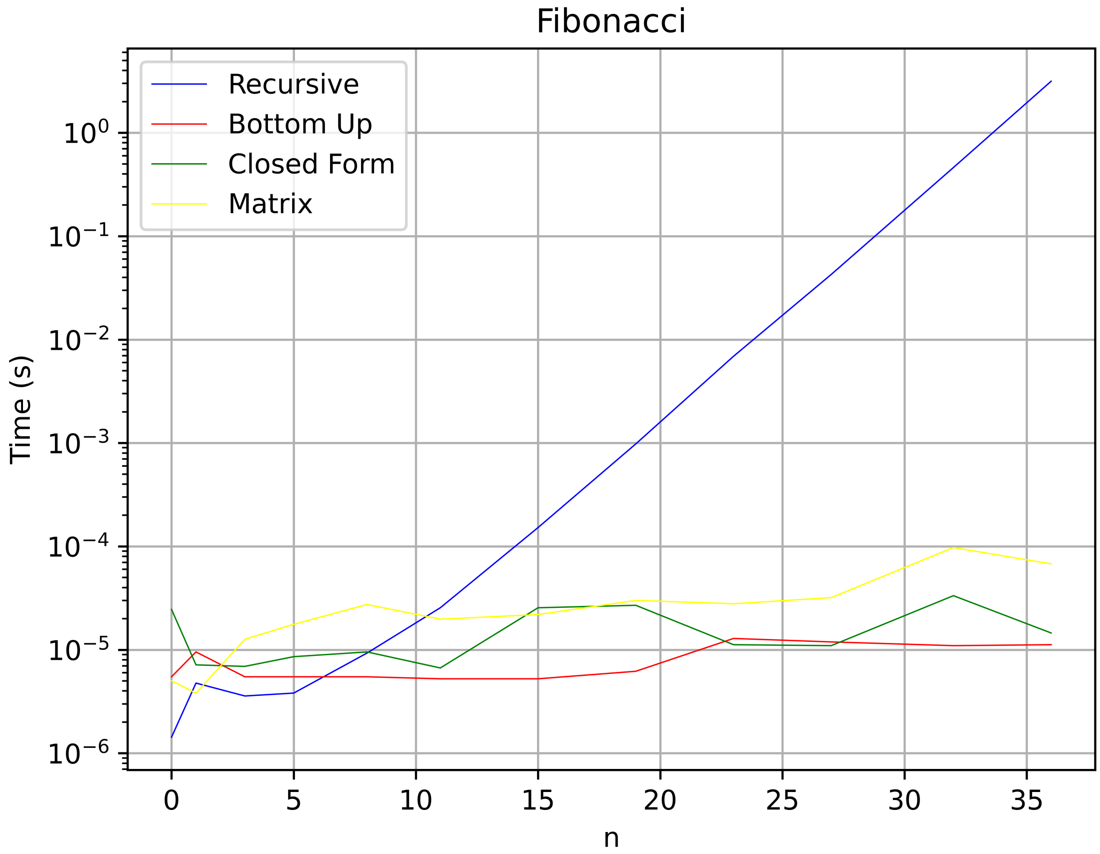
Algorithms and Data Structures: Homework #5

Due on March 9, 2020 at 23:00 PM

Henri Sota

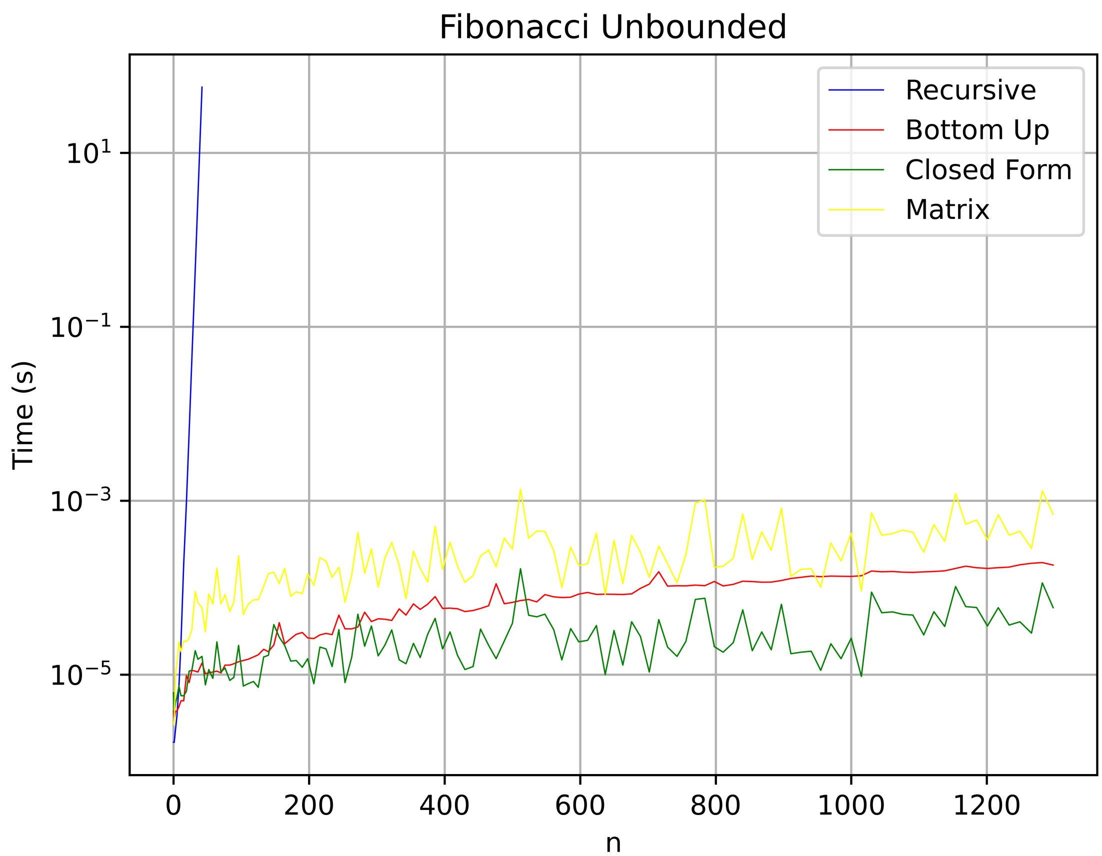
Problem 5.1

d) Plot your results in a line plot, so that the four approaches can be easily compared. Briefly interpret your results.
Hint: Use logarithmic scales for your plot.



In this sample, it can be deduced that Recursive approach takes the highest amount of time to be performed on $n > 10$. Recursive method needs to call itself twice with argument $n-1$ and $n-2$, which needs to perform the same calculations repeatedly. It can be proved that the complexity of the recursive method is exponential and therefore it takes the most amount of time. The other methods take linear time with small coefficient differences.

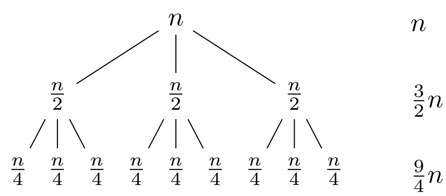
Removing the time limit for the other methods to run until $n \approx 1300$, the runtimes can be seen in the graph below. The matrix method can be seen to be the second most time requiring out of the other 3 methods as it performs extra calculations for all 4 matrix entries. These 3 methods are linear as their graph on a logarithmic scale is approximately a logarithmic line.



Problem 5.2

Consider the problem of multiplying two large integers a and b with n bits each (they are so large in terms of digits that you cannot store them in any basic data type like long long int or similar). You can assume that addition, substraction, and bit shifting can be done in linear time, i.e., in $\Theta(n)$.

d) Solve the recurrence in subpoint (c) using the recursion tree method.



Height of the tree is $\log_2 n$ as the size of each subproblem is half of the original one. Using this information, an exact general formula can be prouced:

$$\sum_{k=0}^{\log_2 n} \frac{3^k}{2^k} n = n \sum_{k=0}^{\log_2 n} (\frac{3}{2})^k$$

The ratio between each consecutive term is: $\frac{3}{2}$. Using the geometric series formula on this sum:

$$\begin{aligned} n \sum_{k=0}^{\log_2 n} (\frac{3}{2})^k &= n \frac{1 - (\frac{3}{2})^k}{1 - \frac{3}{2}} \\ &= n \frac{3^k}{2^{k-1}} - 2n \\ &= n \frac{3^{\log_2 n}}{2^{\log_2 n - 1}} - 2n \end{aligned}$$

By letting $n \to \infty$, the difference between the upper part of the quotient and the lower part of the quotient in the formula is really small and we can combine both of the powers into one:

$$\begin{aligned} n \frac{3^{\log_2 n}}{2^{\log_2 n - 1}} - 2n &= n(\frac{3}{2})^{\log_2} \\ &= n \cdot n^{\log_2 \frac{3}{2}} \\ &= n \cdot n^{\log_2 3 - \log_2 2} \\ &= n \cdot n^{\log_2 3 - 1} \\ &\approx n \cdot n^{0.58} \\ &\approx n^{1.58} \end{aligned}$$

As a result, the time complexity solved using the recursion tree is $\Theta(n^{\log_2 3})$