CS222: Assignment 2 - Fibonacci numbers using recursion and repeated squaring of a matrix

1. Submission deadline: Monday, 16 January at 3 pm.
2. Take $n : 1 \leq n \leq N$. N is a number that depends on your computer's capability. Take N to be at least 40.
3. You can ignore a few initial values of n if they skew the graph.
4. Read, solve and understand Exercise 0.4 of 'Algorithms' by Dasgupta, Papadimitriou, Vazirani. That will help you with this assignment.
5. Follow good coding practices to gain more marks.
6. No copying among the students or from the Internet or any other source.
7. The assignment can be submitted in groups of size ≤ 2.
8. Submit two `.cpp` files and one `.pdf` file.
9. Write the names and roll numbers of the students at the top of each file.
10. The files should be called

    `fibonacci_recursive_firstRollNumber_secondRollNumber.cpp`,

    `fibonacci_repeated_squaring_firstRollNumber_secondRollNumber.cpp`,

    `fibonacci_firstRollNumber_secondRollNumber.pdf`.
11. The pdf should contain the output obtained when each program was run, the line graphs and the answers to the questions asked.

Recall the Fibonacci series:

$$\begin{aligned} F_0 &= 0, \\ F_1 &= 1, \\ F_n &= F_{n-1} + F_{n-2}, \qquad \forall n \geq 2. \end{aligned}$$

1. Implement a *recursive* function that computes the nth Fibonacci number F_n.

   In a line graph, map

   1. $n, \log(F_n)$ and
   2. $n, \log(T(n))$, where $T(n)$ is the time taken to compute F_n.

   Questions:

   1. Conclude that the Fibonacci series and the time taken grows exponentially.

2. What are the slopes of the two lines?
3. Make a guess about F_n as a function of n.
4. Make a guess about $T(n)$ as a function of n.

2. Implement a function that computes the nth Fibonacci number F_n by repeatedly squaring the matrix:

$$\begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}$$

Compute the first N numbers in the Fibonacci sequence. I.e. for each $n \leq N$, call this repeated squaring function separately.

Do the theoretical analysis of your function: Let $M(n)$ be the time complexity of multiplying two integers of n bits. What is the time complexity of your function in terms of $M(n)$?